Oracle® Enterprise Session Border Controller

ACLI Configuration Guide
Release E-CZ7.1.0
Formerly Net-Net Enterprise Session Director

January 2017

ORACLE

Notices

Contents

About this Guide

This guide provides information about:

- Loading the system software image and establishing operating parameters
- Configuring all components and functionality of the Oracle Enterprise Session Border Controller
- Using the features and abilities of the Oracle Enterprise Session Border Controller

Documentation Set

The following table describes the documents included in this release.

Document Name	Document Description
ACLI Configuration Guide	Contains information about the installation, configuration, and administration of the Oracle Enterprise Session Border Controller.
Web GUI Users Guide	Contains information about using the tools and features of the Oracle Enterprise Session Border Controller Web GUI.
Release Notes	Contains information about this release, including platform support, new features, caveats, known issues, and limitations.

Related Documentation

The following table describes related documentation for this release.

Document Name	Document Description
Acme Packet 3820 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3820 system.
Acme Packet 4500 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4500 system.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Oracle Enterprise Session Border Controller.
ACLI Reference Guide	Contains explanations of how to use the ACLI, with alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Acme Packet's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about accounting support, including details about RADIUS accounting.

Document Name	Document Description
HDR Resource Guide	Contains information about the Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about support for the Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Enterprise Session Border Controller family of products.

Revision History

The following table describes updates to this guide.

Date	Revision	Description
July 2014	1.00	• Initial Release • Corrects errors in software based transcoding section
August 2014	1.01	Corrects errors in the platform support topic.
September 2014	1.02	Removes the "Session Replication Recording" section. This function is not supported.
November 2014	1.03	• Updates **restriction-mask** parameter with maximum value of 128 to accommodate for IPv6 address masking • Self-signed certificates are allowed only for MSRP connections.
December 2014	1.04	• Updated trust-me references. • Updates Session Agent Ping Message Formatting Configuration task and Session Agent Ping Message Formatting Configuration topic to reflect valid descriptions for the **session-agent** > **ping-to-user-part** and **session-agent** > **ping-from-user-part** parameters. • Updates Denial of Service feature group.
March 2015	1.05	• Updates the timezone-set command to note that the user can use CTRL-D to exit this command without completing it. • Updates the code block examples in the "Configure a Session Border Controller (SBC) Behind a Network Translation (NAT) Device Option" topic to include the = character between +HeaderNatPrivateSipIftp and the IP address. • Removes KPML from the Events section of the table in the "Subscriptions Report" section. • Removes KPML from the Notable Event section of the table in the "Notable Events Report" section.
May 2015	1.06	• Removes Hyper-V support. • Updates minimum CPU cores requirements. • Notes that the Web GUI supports only IPv4. • Adds the "NTP Synchronization" section.
September 2015	1.07	• Removes reference to unsupported SRR feature.

Date	Revision	Description
		• Removes the Surrogate Registration chapter. This feature is not supported in E series software. • Removes ENUM from the "Compound Key LRT Lookup" topic because ENUM does not support compound lookup keys.
October 2015	1.08	• Updates Session Agent Minimum Reserved Bandwidth topic with limitation to Acme Packet 3820 and 4500
January 2016	1.09	• Removes the FIPS compliance section.
March 2016	1.10	• Corrects the explanation of the tos-value setting. • Adds requirement that any media interface configured with comm-monitor must also belong to a realm. • Corrects description of parameter for source realm selection in multi-stage routing. • Adds the Request URI Construction as Forwarded to SAG-member Session Agent feature description to the Session Routing and Load Balancing Chapter • Renames HIP section in Network Interfaces to Administrative Applications Over Media Interfaces. Corrects description regarding firewalls and ports. Adds gateway requirement and overlapping subnet advisory. • Minor edits to Network Interfaces conceptual and Configuration sections. • Updates HIP Address Configuration for clarity. Adds SSH parameter. Adds gateway requirement and overlapping subnet advisory. • Removes Source-based Routing section from System Configuration chapter.
April 2016	1.11	• Removes the topic- Manually setting timezone • Corrects call flow diagrams for refer-initiated transfers
June 2016	1.12	• Added a note in the section Tunnel mode vs Transport mode under IPSec support • Edited description for IPSec support
September 2016	1.13	• Adds new scenario under Fax Transcoding topic
December 2016	1.14	• Updates Conditional Codec policies
January 2016	1.15	• Adds a note to "Configurable Alarm Thresholds and Traps" regarding a change to critical memory alarm threshold functionality. • Updates Ingress Policy in Transcoding

1

Oracle Enterprise Session Border Controller Basics

This chapter introduces some basic concepts that apply to the key features and abilities of your Oracle Enterprise Session Border Controller. It is necessary that you understand the information included in this chapter to comprehend the ways to configure your Oracle Enterprise Session Border Controller. This chapter only provides a high level overview of some important Oracle Enterprise Session Border Controller concepts.

Oracle Enterprise Session Border Controller Description

The Oracle Enterprise Session Border Controller (E-SBC) connects disparate Internet Protocol (IP) communications networks while mitigating security threats, curing interoperability problems, and ensuring reliable communications. The E-SBC protects and controls real-time voice, video, and Unified Communications (UC) as they traverse IP network borders.

Overview

Available in software and appliance configurations, the E-SBC is highly scalable and includes an industry-leading feature set.

- Strong security. As the E-SBC protects IP telephony and UC infrastructure, services, and applications, it also ensures confidentiality, integrity, and availability. The E-SBC protects against fraud, service theft, malicious attacks, system overloads, and other events that affect service.
- Easy interoperability. The E-SBC provides extensive signaling and media control features to help businesses overcome interoperability challenges that commonly occur when interfacing with public IP network services. The E-SBC also performs protocol interworking and dial plan management for integration with legacy systems.
- Assured reliability. The E-SBC ensures Public Switched Telephone Networks (PSTN)-like availability and service quality for IP communications. The E-SBC enforces service quality, balances loads across trunks, and reroutes sessions around interface disruptions to optimize network performance, circumvents equipment and facility problems, and ensures business continuity.

Functions and Modes

Businesses install the E-SBC at Session Initiation Protocol (SIP) network borders, where enterprise communications systems interface with public network services and where disparate multi-vendor systems must be managed.

Customers use the E-SBC to:

- Connect to SIP trunking services and the Internet
- Access communications services

- Communicate securely with remote workers
- Manage sessions across a multi-vendor UC environment
- Connect contact center locations and Business Process Outsourcing (BPO) services

What Is a Realm

A realm is a logical way of identifying a domain, a network, a collection of networks, or a set of addresses. Realms are used when a Oracle Enterprise Session Border Controller communicates with multiple network elements over a shared intermediate connection. Defining realms allows flows to pass through a connection point between two networks.

From an external perspective, a realm is a collection of systems that generates real-time interactive communication sessions comprised of signaling messages and media flows, or a group of multiple networks containing these systems. These systems may be session agents such as call agents, softswtiches, SIP proxies, H.323 gatekeepers, IP PBXs, etc., that can be defined by IPv4 addresses. These systems can also be IP endpoints such as SIP phones, IADs, MTAs, media gateways, etc.

From an internal perspective, a realm is associated with Oracle Enterprise Session Border Controller configurations to define interfaces and resources in a logical way. Realms are used to support policies that control the collection of systems or networks that generate media sessions. Realms are referenced by other configuration elements in order to support this functionality across the protocol the Oracle Enterprise Session Border Controller supports and to make routing decisions.

Nested Realms

Nested Realms is a Oracle Enterprise Session Border Controller feature that supports hierarchical realm groups. One or more realms may be nested within higher order realms. Realms and sub-realms may be created for media and bandwidth management purposes. This feature supports:

- Separation of signaling & media on unique network interfaces
- Signaling channel aggregation for Hosted IP Services applications
- Configuration scalability
- Per-realm media scalability beyond single physical interface capacity
- Nested bandwidth admission control policies

What Is a Session Agent

A session agent defines an internal signaling endpoint. It is an internal next hop signaling entity that applies traffic shaping attributes to flows. For each session agent, concurrent session capacity and rate attributes can be defined. Service elements such as gateways, softswitches, and gatekeepers are defined automatically within the Oracle Enterprise Session Border Controller as session agents. The Oracle Enterprise Session Border Controller can also provide load balancing across the defined session agents.

SIP session agents

SIP session agents can include the following:

- Softswitches
- SIP proxies
- Application servers
- SIP gateways

H.323 session agents

H.323 session agents can include the following:

- gatekeepers
- gateways
- MCUs

Why You Need Session Agents

You can use session agents to describe next or previous hops. You can also define and identify preferred carriers to use for traffic coming from session agents. This set of carriers is matched against the local policy for requests coming from the session agent. Constraints can also be set for specific hops.

In addition to functioning as a logical next hop for a signaling message, session agents can provide information regarding next hops or previous hops for SIP packets, including providing a list of equivalent next hops.

How to Use Session Agents

You can use session agents and session agent groups (along with local policies) to define session routing for SIP and H.323 traffic. You can associate a realm with a session agent to identify the realm for sessions coming from or going to the session agent.

What is a Session Agent Group

A session agent group contains individual session agents bundled together. A SAG indicates that its members are logically equivalent and can be used interchangeably. This allows for the creation of constructs like hunt groups for application servers or gateways. Session agent groups also assist in load balancing among session agents.

Session agent groups can be logically equivalent to the following:

- Application server cluster
- Media gateway cluster
- Softswitch redundant pair
- SIP proxy redundant pair
- Gatekeeper redundant pair

High Availability

High Availability (HA) is a network configuration used to ensure that planned and unplanned outages do not disrupt service. In an HA configuration, Oracle Enterprise Session Border Controllers (E-SBC) are deployed in a pair to deliver continuous high availability for interactive communication services. Two E-SBCs operating in this way are called an HA node. The HA node design ensures that no stable call is dropped in the event of an outage.

In an HA node, one E-SBC operates in the active mode and the other E-SBCoperates in the standby mode.

- Active. The active member of the HA node is the system actively processing signal and media traffic. The active member continuously monitors itself for internal process and IP connectivity health. If the active member detects a condition that can interrupt or degrade service, it hands over its role as the active member of the HA node to the standby member.
- Standby. The standby member of the HA node is the backup system. The standby member is fully synchronized with active member's session status, but it does not actively process signal and media traffic. The standby member monitors the status of the active member and it can assume the active role without the active system having to instruct it to do so. When the standby system assumes the active role, it notifies network management using an SNMP trap.

The E-SBC establishes active and standby roles in the following ways.

- If an E-SBC boots up and is alone in the network, it is automatically the active system. If you pair a second E-SBC with the first one to form an HA node, the second system automatically establishes itself as the standby.
- If both E-SBCs in the HA node boot up at the same time, they negotiate with each other for the active role. If both systems have perfect health, then the E-SBC with the lowest HA rear interface IPv4 address becomes the active E-SBC. The E-SBC with the higher HA rear interface IPv4 address becomes the standby E-SBC.

If the rear physical link between the twoE-SBCs is unresponsive during boot up or operation, both will attempt to become the active E-SBC. In this circumstance, processing does not work properly.

The standby E-SBC assumes the active role when:

- it does not receive a checkpoint message from the active E-SBC for a certain period of time.
- it determines that the active E-SBC health score declined to an unacceptable level.
- the active E-SBC relinquishes the active role.

To produce a seamless switch over from one E-SBC to the other, the HA node members share their virtual MAC and virtual IP addresses for the media interfaces in a way that is similar to Virtual Router Redundancy Protocol (VRRP). Sharing these addresses eliminates the possibility that the MAC address and the IPv4 address set on one E-SBC in an HA node will be a single point of failure. Within the HA node, the E-SBCs advertise their current state and health to one another in checkpointing messages to apprise each one of the other one's status. Using the Oracle HA protocol, theE-SBCs communicate with UDP messages sent out and received on the rear interfaces. During a switch over, the standby E-SBC sends out an ARP request using the virtual MAC address to establish that MAC address on another physical port within the Ethernet switch. To the upstream router, the MAC address and IP address are still alive. Existing sessions continue uninterrupted.

2

Supported Platforms

Platform Support

The following platforms support the E-CZ7.1.0 release.

- Oracle: AP3820 and AP4500
- Virtual Machine Edition: VMWare

Release Image File Names

Use the following files for a new deployment.

Oracle Hardware

- Image: nnECZ710.64.bz for the AP4500 and nnECZ710.32 .bz for the AP 3820
- Boot loader: July 2013 or newer

Virtual Machines

- VMWare: nnECZ710.64-img-bin.ova

Upgrade Image File Names

Use the following files to upgrade virtual machine deployments.

- Image: nnECZ710.64.bz
- Boot loader: nnECZ710.64.boot

Virtual Machine (VM) Edition

VMware virtual machine images are deployed as Open Virtualization Format (OVA) files that provide a file directory, an OVF template, and a virtual disk image that contains all required Oracle Enterprise Session Border Controller software. The VM software supports Oracle VM, VMware VSphere, and ESXi 5.5 Hypervisor.

Oracle recommends using VMware ESXi because it supports network booting and Dynamic Host Configuration Protocol (DHCP). The hypervisor supports up to 250 SIP audio sessions per VM.

For installation instructions, see "Creating and Deploying VMware Net-Net OS VM Edition."

Build Images for Virtual Machines

Use the following images for installing new virtual machines.

- VMware. Use the nnECZ710-img-bin.ova build image, which is delivered as an Open Virtualization Archive (.ova) file.
- Windows 2008 R2. Use the nnECZ710-img-bin.vhd build image, which is delivered as a virtual hard disk (.vhd) file.
- Evaluation. The evaluation version of the nnECZ710-img-bin.ova and nnECZ710-img-bin.vhd build images supports a 90-day trial period.

Minimum Virtual Machine Resources

Each VM instance, regardless of the virtualization environment (VMware or Windows) requires the following minimum allocation or network resources.

- CPU cores: 4
- Memory: 4GB
- Hard drive storage: 40GB
- 64-bit application
- Interfaces: 8 recommended (you can use fewer)

Virtual Machine (VM) Configuration

Use the Oracle Enterprise Session Border ControllerACLI to perform the following configuration, which is required for all VM operations. Oracle recommends performing the following operations immediately after creating and deploying the VM.

- Set boot parameters
- Format the VM hard disk

3

Software Editions

Overview

This chapter provides an overview of common features and functionality provided by the VM Edition.

Common Functionality

The VM Edition includes the following functionality that is common across all Oracle Enterprise Session Border Controller platforms.

- ACLI structure, syntax and process (as well as most commands)
- Net-Net Central support
- Net-Net ESD configuration
- Security certificates loaded and stored
- Peering and access models
- SIP trunking
- High availability

Denial of Service (DoS) Protection

Support for DoS protection in the VM Edition of the Oracle Enterprise Session Border Controller differs from the Oracle Hardware Platforms Edition because of the absence of Oracle network interface hardware. In the VM Edition. DoS protection is implemented in the software and consumes CPU cycles when responding to attacks.

The VM Edition handles media packet fragments differently from the Oracle Hardware Platforms Edition by processing the fragments in the data path rather than in the host application code. Protection against fragment attacks is ensured because fragments are never kept more than 5 ms.

Denial of Service (DoS) Calculations

DoS provisioning is accomplished in the media-manager configuration mode. Three new parameters supported in VM Edition define DoS thresholds.

- max-trusted-packet-rate specifies the maximum trusted packet rate in packets/second
- max-untrusted-packet-rate specifies the maximum untrusted packet rate in packets/second
- max-arp-packet-rate specifies the maximum ARP packet rate in packets/second

While the configured rate is expressed as packets/second, the actual rate is measured as packets/millisecond. The following illustration shows configured rates and actual rates.

```
                            Configured       Actual
                          Rate             Rate
max-trusted-packet-rate   3200 pkts/sec    3 pkts/ms
max-untrusted-packet-rate 1700 pkts/sec    1 pkt/ms
max-arp-packet-rate       1200 pkts/sec    1 pkt/ms
```

Displays for show commands, such as show datapath DOS settings, report the millisecond-based actual rate, leading to the apparent discrepancy between the configured rate and the displayed rate as shown in the following illustration.

```
                           Configured    Actual          Displayed
                           Rate          Rate      Rate
max-trusted-packet-rate    3200 pkts/sec 3 pkts/ms    3000 pkts/sec
max-untrusted-packet-rate  1700 pkts/sec 1 pkt/ms     1000 pkts/sec
```

Ingress Queues

The ingress packets destined for the host are placed in one of four queues:

- untrusted
- trusted
- ARP request
- ARP reply

Events such as latching and RFC2833 translation are placed in a fifth queue. The event queue has the highest priority and is emptied for each iteration, which ensures control traffic is not blocked under DoS attacks.

Net-Net ESD, Server Edition supports a maximum of 8000 trusted endpoints. Currently, when the trusted queue is full, the next endpoint coming in enters the untrusted queue. This is reported in the output of the show acl trusted as Trusted Entries not allocated due to ACL constraints:.

Denial of Service (DoS) Configuration Defaults

The following example shows default DoS values in the media-manager configuration.

```
media-manager
max-signaling-bandwidth 332000
max-untrusted-signaling 100
min-untrusted-signaling 30
app-signaling-bandwidth 0
tolerance-window 30
arp-msg-bandwidth 32000
```

Based on the default values, the system calculates the trusted, untrusted, ARP, and events packets per second.

Use the show sw-datapath DOS settings command to display these values.

```
# show sw-datapath DOS settings
Queue Size PPS
ARP reply 375 250
ARP request 375 250
Trusted 210 142
Untrusted 300 200
Event 8192 9000
```

Packet Trace PCAP

The VM Edition packet tracing support differs from the other Oracle Enterprise Session Border Controller platforms such as Net-Net 3800 and Net-Net 4500. When enabled, packets are captured that meet specific criteria. The packets are logged into a file in the /opt/traces directory in a PCAP-formatted format as well as being displayed to the ACLI session from which the capture was executed.

You can enable or disable packet capture on the Net-Net ESD. The default filter uses port 5060 on the specified interface to capture both ingress and egress ICMP traffic.

Packet Capture Initiation

From the command line, you can initiate a live packet capture session to view packet traffic on your network. For example, you might want to confirm the network configuration or to perform troubleshooting.

During a packet capture session, the system creates a set of .pcap files in the /opt/traces directory. If the /opt/traces directory contains files when you run the packet-trace command, the system prompts you to either remove or keep the existing files before running the command. The following table describes the system behavior for both options.

Option	Result	Packet Trace Command Behavior
Yes	Removes all existing files.	The system captures up to 25 new .pcap files. During the session, the system rotates the files in the /opt/traces directory by size. For example, the system keeps the last 25 files and rotates them when they reach 100 MB
No	Keeps all existing files.	• If the /opt/traces directory contains 25 .pcap files, the system cannot add more files to the directory or overwrite the existing files. • If the /opt/traces directory contains fewer than 25 .pcap files, the system can add new files to the directory up to the 25 file limit. For example, if the /opt/traces directory contains 10 existing files, the system can add up to 15 new files.

Use the packet-trace [local | remote] [start | stop] [<network-interface>] command to initiate a packet capture session.

ACLI Command Support

This section describes common ACLI commands supported only on the VM Edition. Because the VM Edition is a software-only product, you may find that ACLI commands relating to hardware components are not supported or differ in syntax and output.

For more information about using the ACLI Show commands in this section, see the *Net-Net® 4000 Maintenance and Troubleshooting Guide*.

Commands Related to Hardware

The following commands relating to hardware components are not supported:

- show prom-info
- fragment-msg-bandwidth
- show qos

4

Getting Started

Prior to configuring your Oracle Enterprise Session Border Controller for service, we recommend that you review the information and procedures in this chapter.

This chapter offers information that will help you:

- Review hardware installation procedures
- Connect to your Oracle Enterprise Session Border Controller using a console connection, Telnet, or SSH (secure shell)
- Become familiar with the Oracle Enterprise Session Border Controller's boot parameters and how to change them if needed
- Obtain, add, and delete Oracle Enterprise Session Border Controller software licenses
- Load and activate a Oracle Enterprise Session Border Controller software image
- Choose a configuration mechanism: ACLI, Oracle Communcations Session Element Manager or ACP/XML
- Enable RADIUS authentication
- Customize your login banner

Enterprise Software Licensing

A valid license is required for each Oracle Enterprise Session Border Controller software component that you want to use. To obtain a valid license, you provide the serial number from the system to Oracle and request a license key.

The following guidelines apply to Oracle Enterprise Session Border Controller software licenses.

- Each license is bound to a specific Oracle Enterprise Session Border Controller by serial number.
- Oracle does not allow transferring a license from one Oracle Enterprise Session Border Controller to another.
- The system supports multiple licenses that are active simultaneously on the same Oracle Enterprise Session Border Controller.
- If a feature is covered by more than one license, the latest expiration date applies.
- The system can activate and deactivate a license in real time.
- A license is fully extensible and upgradable.
- If a component is unlicensed, the system does not display the interface and configuration parameters for the component.
- Software licenses are aggregate. When a new license is added to the original license set, the related capacity, protocol, or interface becomes part of the functionality that you can configure and deploy. For example, if your original license for session capacity is 1,000 and you add a new license for 3,000 sessions, the new total session capacity is 4,000.

For more information, see "Licensing Document" on the Oracle Software Delivery Cloud website in the directory with the software image download file.

Obtain a License

Before You Begin

Download the software for the component that you want to license.

A valid license is required for each Oracle Enterprise Session Border Controller software component that you want to use. To obtain a valid license, provide the serial number from the system to Oracle and request a license key.

1. On your system, do one of the following to locate the serial number.
   - Hardware. Use the show prom-info mainboard command.
   - VMWare. Use the show version boot command.
2. Go to http://www.oracle.com/us/support/licensecodes/index.html and request a license key.
3. Add the license using the add command in Configuration mode.
4. Verify the license using the show features command in the Privileged mode.

Trial License

Oracle offers a trial license for software components to allow testing a feature before deployment.

A trial license is active for specified period of time. When the trial license expires, the feature stops responding and the system removes the configuration selections. To continue using the feature you must obtain a license. For more information about licensing, contact your Oracle sales representative or technical support representative.

License Expiration

When a license expires, you are no longer able to use the features associated with it. The Oracle Enterprise Session Border Controller automatically disables all associated processes.

To avoid a license unexpectedly expiring and therefore potentially disrupting service, we recommend that you track expiration dates and renew licenses well in advance of expiration.

Expired licenses appear in your Oracle Enterprise Session Border Controller ACLI displays until you delete them, though you cannot use the features associated with them. Deleting an expired license requires that you take the same steps as you do for deleting a valid one.

Viewing Licenses

There are two ways to view licenses in the ACLI.

- You can use the show features command at the main ACLI user prompt.

```
ACMEPACKET# show features
Total session capacity: 2250
Enabled protocols: SIP, MGCP, H.323, IWF
Enabled features: ACP
ACMEPACKET#
```

- Within the license menu, use the show command to see all licenses with detailed information.

```
ACMEPACKET(license)# show
License #1: 2000 sessions, SIP, MGCP, ACP
                no expiration
                installed at 12:34:42 APR 01 2005
License #2: H323
                expired at 23:59:59 APR 08 2005
                installed at 12:35:43 APR 01 2005
License #3: 250 sessions, IWF
                expires at 23:59:59 APR 28 2005
                installed at 12:36:44 APR 01 2005
```

```
License #4: QOS
                starts at 00:00:00 APR 08 2004
                expires at 23:59:59 OCT 27 2005
                installed at 12:37:45 APR 01 2005
Total session capacity: 2250
ACMEPACKET(license)#
```

Licensing Information for the Acme Packet 3800

Although all features currently available on the Acme Packet 4000 series of products are available on the Acme Packet 3800, you will see some minor changes in licensing when using this newest addition to the Acme Packet family of products. These changes include:

- Session capacity limits
- Finer session capacity granularity
- Denial of Service
- Software TLS

Session Capacity and Your Net-Net 3800

The Net-Net 3800 supports a maximum limit of 8000 concurrent sessions. The following values are the session capacity values you can license for the Net-Net 3800:

- 25
- 50
- 100
- 150
- 250
- 350
- 500
- 1000
- 2000
- 4000
- up to 8000

Additional session capacities may be added at a later date through purchase of sessions in increments of 25, 50 or 100. Session capacity is additive in the Net-Net 3800, meaning the total number of sessions for the system is the sum of all session capacities licensed. The sum total of the licenses cannot exceed 8000 sessions. The Net-Net 3800 strictly enforces this limit.

Granularity and Oversubscription Limits

Only on the Acme Packet 3810, the Oracle Enterprise Session Border Controller uses a 10-to-1 oversubscription limit, meaning that the system allows ten registrations for a single licensed session. The system enforces the limits across all signalling protocols.

An SNMP OID, apSysRegistrationCapacity, supports querying the percentage of used registration capacity. When the percentage approaches the registration capacity limit, an alarm triggers and the Acme Packet 3810 sends an SNMP trap.

- SIP—For SIP, the 10-to-1 ratio limits has possible implications for the SIP registrations cache limiting feature. When you enable that feature, the Oracle Enterprise Session Border Controller rejects new registrations when they exceed the configurable registration cache limit. Likewise, the system can rejects registrations when they exceed the global oversubscription limit. It uses whichever is the lower of the two.

  The Acme Packet 3810 first checks the configurable registrations cache limits. If you have configured this value to be higher than the global oversubscription limits, the Acme Packet 3810 leaves the registration cache limit value intact. However, if registrations go over the global oversubscription limit, the Acme Packet 3810 will reject them, regardless of the cache limit, and the corresponding traps and alarms might not be triggered.

- H.323—The Acme Packet 3800 tracks the number of CallSignalingAddress records as a means of counting registrations. This methods relies on each endpoint having a unique CallSignalingAddress.
- MGCP—Since there can be an unknown number of endpoints registered at once with MGCP, the Acme Packet 3800 uses the count called MGCP Sessions shown in the MGCP statistics display s a way to count the number of registrations. Note that this value is different from the one listed for MGCP media sessions.

SNMP Support for Global Registration Capacity

For the Acme Packet 3800 only, you can use the apSysRegistrationCapacity object to query the percentage of used global registration capacity on your system. This object and corresponding group are now part of the apSystemManagement MIB, ap-smgmt.mib. The OID and its value are also sent as parameters in the apSysMgmtGroupTrap when an alarm condition occurs. The alarm for this condition is SYS_REG_OVER_THRESHOLD with these values: 0x0002003A (hexidecimal) and 131130 (decimal).

The alarm condition depends on whether or not you have set any alarm thresholds for the session type in the system configuration.

- If you have configured them, the thresholds apply to registration capacity. The registration capacity alarm uses the same percentage values and severities for the alarm as those set for the session alarm thresholds.
- If you have not configured them, then the registration capacity alarm triggers at 90%.

The alarm clears when two successive checks, performed once every five seconds, report a value under the threshold.

Denial of Service Feature Group

For the Acme Packet 3800 only, a denial of service (DoS) license now exists. When the DoS license is not present, certain whole configurations and specific parameters within unrestricted configurations related to DoS functionality are not available. You can neither configure them, nor can you see them when you use the ACLI show configuration command.

The table below details the restrictions.

Restricted Configuration Element	Restricted Parameters
access-control	realm-id source-address destination-address application-protocol transport-protocol access average-rate-limit trust-level invalid-signal-threshold maximum-signal-threshold untrusted-signal-threshold deny-period
media-manager	max-signaling-bandwidth max-untrusted-signaling min-untrusted-signaling fragment-msg-bandwidth tolerance-window

Restricted Configuration Element	Restricted Parameters
	arp-msg-bandwidth rtcp-rate-limit
media-profile	average-rate-limit
realm-config	average-rate-limit access-control-trust-level invalid-signal-threshold maximum-signal-threshold untrusted-signal-threshold nat-trust-threshold deny-period
static-flow	average-rate-limit

Software TLS Feature Group

Software TLS is a feature group for the Net-Net 3800 only. It allows for the use of TLS functionality without the presence of an SSM card. If you want to achieve higher capacity for TLS on your Net-Net 3800, you can use the SSM card.

Unlicensed Oracle Enterprise Session Border Controller

When your Oracle Enterprise Session Border Controller arrives, you must obtain a key to activate the licenses for the functionality you want to use. Obtain the license key from Oracle customer support at http://www.oracle.com/us/support/licensecodes/index.html.

If you log onto an unlicensed Oracle Enterprise Session Border Controller, the system displays a warning message that a valid licence is required. Until you enter a valid license, you can configure general system parameters, but not parameters for protocols and features.

Standalone System Licensing

This section shows you how to add licenses and delete them from standalone Oracle Enterprise Session Border Controllers. The process for two systems making up an HA node is different, so follow the procedure relevant to your configuration.

Adding a License to a Standalone System

Once you have obtained a license key, you can add it to your Oracle Enterprise Session Border Controller and activate it.

To add and activate a license on your Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type license and press Enter.

```
ACMEPACKET(system)# license
ACMEPACKET(license)#
```

4. Using the add command and the key generated by Oracle, add the license to your Oracle Enterprise Session Border Controller.

```
ACMEPACKET(license)# add sl25o39pvtqhas4v2r2jc1oaen9e01o21b1dmh3
```

5. You can check that the license has been added by using the ACLI show command within the license configuration.

```
ACMEPACKET(license)# show
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
ACMEPACKET(license)#
```

6. To activate your license, type the activate-config command and press Enter. The Oracle Enterprise Session Border Controller then enables any of the processes that support associated features.

```
ACMEPACKET# activate-config
```

Deleting a License from a Standalone System

You can delete a license from your Oracle Enterprise Session Border Controller, including licenses that have not expired. If you want to delete a license that has not expired, you need to confirm the deletion.

To delete a license from the Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type license and press Enter.

```
ACMEPACKET(system)# license
ACMEPACKET(license)#
```

4. Type the no command and press Enter. A list of possible licenses to delete appears.

```
ACMEPACKET(license)# no
feature:
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
selection:
```

5. Type the number corresponding to the license you want to delete and press Enter.

```
selection:7
```

6. If the license has not expired, you are be asked to confirm the deletion.

```
Delete unexpired license [y/n]?: y
ACMEPACKET(license)#
```

When you show the licenses, the one you deleted should no longer appear on the list.

7. To clear the license from the system, type the activate-config command and press Enter. The Oracle Enterprise Session Border Controller then disables any of the processes that support associated features.

```
ACMEPACKET# activate-config
```

High Availability (HA) Pair Licensing

When adding and deleting licenses, you must perform the procedure across both members of an HA pair during the same service window. The licenses on each peer must be identical. Peers with mismatched licenses may exhibit unexpected behavior.

Adding a License to an HA Node

To add a license to both systems in an HA node, you add your licenses to both systems and reboot both systems when they are in standby mode. The process requires that you carefully confirm system synchronization in between various steps.

This procedure uses the designations Oracle Enterprise Session Border Controller1 as the original active and Oracle Enterprise Session Border Controller2 as the original standby.

To add a license on systems in an HA node:

1. Confirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

   You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 5.

   - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
   - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

     ```
     ACMEPACKET1# display-current-cfg-version
     Current configuration version is 5
     ACMEPACKET1#
     ACMEPACKET2# display-current-cfg-version
     Current configuration version is 5
     ACMEPACKET2#
     ```

   - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

     ```
     ACMEPACKET1# display-running-cfg-version
     Running configuration version is 5
     ACMEPACKET1#
     ACMEPACKET2# display-running-cfg-version
     Running configuration version is 5
     ACMEPACKET2#
     ```

2. Now you can add a license to SBC1. To begin, type configure terminal and press Enter.

   ```
   ACMEPACKET1# configure terminal
   ACMEPACKET1(configure)#
   ```

3. Type system and press Enter.

   ```
   ACMEPACKET1(configure)# system
   ACMEPACKET1(system)#
   ```

4. Type license and press Enter.

   ```
   ACMEPACKET1(system)# license
   ACMEPACKET1(license)#
   ```

5. Using the add command and the key generated by Oracle, add the license to your Oracle Enterprise Session Border Controller.

```
ACMEPACKET1(license)# add sjkl4i45987p43hh0938hnhjlaie10983
```

6. You can check that the license has been added by using the ACLI show command within the license configuration.

```
ACMEPACKET1(license)# show
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
ACMEPACKET1(license)#
```

7. Repeat typing exit, pressing Enter after each entry, until you reach the main Superuser prompt.

```
ACMEPACKET1(license)# exit
ACMEPACKET1(system)# exit
ACMEPACKET1(configure)# exit
ACMEPACKET1#
```

8. Repeat steps 2 through 7 on SBC2.

```
ACMEPACKET2(license)# add [License for SBC2]
```

   At the end of this step, licenses are installed and verified on both SBC1 and SBC2.

9. Return to SBC1 and type the save-config command and press Enter.

```
ACMEPACKET1# save-config
```

10. Type the activate-config command and press Enter. The Oracle Enterprise Session Border Controller then enables any of the processes that support associated features.

```
ACMEPACKET1# activate-config
```

11. Confirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 6.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 6
      ACMEPACKET1#
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 6
      ACMEPACKET2#
      ```

    - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-running-cfg-version
      Running configuration version is 6
      ACMEPACKET1#
      ACMEPACKET2# display-running-cfg-version
      ```

```
Running configuration version is 6
ACMEPACKET2#
```

12. Execute the ACLI show health command to make sure that all processes are synchronized.
13. Return to SBC2 and execute the reboot command.

```
ACMEPACKET2# reboot
```

14. Reconfirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 6.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 6
      ACMEPACKET1#
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 6
      ACMEPACKET2#
      ```

    - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-running-cfg-version
      Running configuration version is 6
      ACMEPACKET1#
      ACMEPACKET2# display-running-cfg-version
      Running configuration version is 6
      ACMEPACKET2#
      ```

15. Trigger a switchover between the two systems in the HA node so the originally standby system assumes the active role. This means that the standby system will transition to active.

```
ACMEPACKET1# notify berpd force
```

16. Wait for Oracle Enterprise Session Border Controller2 to transition to the active state. Confirm that it is in the active state by using the ACLI show health command.
17. Reconfirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must also make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 6.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller2, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller1 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller2.

      ```
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 6
      ACMEPACKET2#
      ACMEPACKET1# display-current-cfg-version
      ```

```
Current configuration version is 6
ACMEPACKET1#
```

- On Oracle Enterprise Session Border Controller2, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller1 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller2.

```
ACMEPACKET2# display-running-cfg-version
Running configuration version is 6
ACMEPACKET2#
ACMEPACKET1# display-running-cfg-version
Running configuration version is 6
ACMEPACKET1#
```

18. Return to SBC1 and execute the reboot command.

```
ACMEPACKET1# reboot
```

19. Wait for SBC1 to complete rebooting. When finished, execute the ACLI show health command to make sure that all processes are synchronized.
    At this point both SBCs should be synchronized and contain the same license configuration.
20. If desired, trigger a switchover between the two systems in the HA node so the originally active system (SBC1) assumes the active role.

```
ACMEPACKET2# notify berpd force
```

    At this point both SBCs should be synchronized and contain the same license configuration.

Deleting a License from an HA Node

To delete a license from both systems in an HA node, you remove your licenses from both systems and reboot both systems when they are in standby mode. The process requires that you carefully confirm system synchronization in between various steps.

Note: Licenses should be deleted on both nodes during the same service window.

This procedure uses the designations Oracle Enterprise Session Border Controller1 as the original active and Oracle Enterprise Session Border Controller2 as the original standby.

To delete a license from systems in an HA node:

1. Confirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

   You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 7.

   - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
   - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

```
ACMEPACKET1# display-current-cfg-version
Current configuration version is 7
ACMEPACKET1#
ACMEPACKET2# display-current-cfg-version
Current configuration version is 7
ACMEPACKET2#
```

   - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border

Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

```
ACMEPACKET1# display-running-cfg-version
Running configuration version is 7
ACMEPACKET1#
ACMEPACKET2# display-running-cfg-version
Running configuration version is 7
ACMEPACKET2#
```

2. Now you can delete a license from SBC1. To begin, type configure terminal and press Enter.

```
ACMEPACKET1# configure terminal
ACMEPACKET1(configure)#
```

3. Type system and press Enter.

```
ACMEPACKET1(configure)# system
```

4. Type license and press Enter.

```
ACMEPACKET1(system)# license
ACMEPACKET1(license)#
```

5. Type the no command and press Enter. A list of possible license to delete appears.

```
ACMEPACKET1(license)# no
feature:
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
selection:
```

6. Type the number corresponding to the license you want to delete and press Enter.

```
selection:7
```

7. If the license has not expired, you are be asked to confirm the deletion.

```
Delete unexpired license [y/n]?: y
ACMEPACKET1(license)#
```

When you show the licenses, the one you deleted should no longer appear on the list.

8. Repeat typing exit, pressing Enter after each entry, until you reach the main Superuser prompt.

```
ACMEPACKET1(license)# exit
ACMEPACKET1(system)# exit
ACMEPACKET1(configure)# exit
ACMEPACKET1#
```

9. Repeat steps 2 through 8 on SBC2.

```
ACMEPACKET2(license)# no
feature:
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
selection:
```

At the end of this step, licenses are removed and verified on both SBC1 and SBC2.

10. Return to SBC1 and type the save-config command and press Enter.

```
ACMEPACKET1# save-config
```

11. Type the activate-config command and press Enter.

```
ACMEPACKET1# activate-config
```

12. Confirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 5.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET1#
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET2#
      ```

    - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-running-cfg-version
      Running configuration version is 7
      ACMEPACKET1#
      ACMEPACKET2# display-running-cfg-version
      Running configuration version is 7
      ACMEPACKET2#
      ```

13. Execute the ACLI show health command to make sure that all processes are synchronized.
14. Return to SBC2 and execute the reboot command.

```
ACMEPACKET2# reboot
```

15. Reconfirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 7.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

      ```
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET1#
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET2#
      ```

    - On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border

Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

```
ACMEPACKET1# display-running-cfg-version
Running configuration version is 7
ACMEPACKET1#
ACMEPACKET2# display-running-cfg-version
Running configuration version is 7
ACMEPACKET2#
```

16. Trigger a switchover between the two systems in the HA node so the originally standby system assumes the active role. This means that the originally standby system will transition to active.

```
ACMEPACKET1# notify berpd force
```

17. Wait for Oracle Enterprise Session Border Controller2 to transition to the active state. Confirm that it is in the active state by using the ACLI show health command.
18. Reconfirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 are synchronized.

    You must also make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the examples below, all of the configuration versions are 8.

    - On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.
    - On Oracle Enterprise Session Border Controller2, show the current configuration version by using the ACLI display-current-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller1 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller2.

      ```
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET2#
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET1#
      ```

    - On Oracle Enterprise Session Border Controller2, show the running configuration version by using the ACLI display-running-cfg-version command. Then use the same command on Oracle Enterprise Session Border Controller1 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller2.

      ```
      ACMEPACKET2# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET2#
      ACMEPACKET1# display-current-cfg-version
      Current configuration version is 7
      ACMEPACKET1#
      ```

19. Return to SBC1 and execute the reboot command.

```
ACMEPACKET1# reboot
```

20. Wait for SBC1 to complete rebooting. When finished, execute the ACLI show health command to make sure that all processes are synchronized.
21. If desired, trigger a switchover between the two systems in the HA node so the originally active system (SBC1) assumes the active role.

```
ACMEPACKET2# notify berpd force
```

At this point both SBCs should be synchronized and contain the same license configuration.

Download the Software

To get the Oracle Enterprise Session Border Controller software, go to the Oracle Cloud Software Delivery website. With an account and a product license, you can download the software.

Before You Begin

Confirm that you have an account with Oracle and a license for the product that you want to download.

Download the Media Pack, software, and License Document for the product and platform that you want. In step 8, Oracle recommends that you view the Readme before attempting the software download.

1. Go to Oracle Cloud Software Delivery at https://edelivery.oracle.com/, and sign in.
2. On the Terms and Restrictions page, in the Oracle Trial License Agreement section, select one of the following:
   - If you want a trial license, select **Yes**.
   - If you have a license, select **Or**.
3. On the Terms and Restrictions page, in the Export Restrictions section, select **Yes**.
4. Click **Continue**.
5. On the Media Pack Search page, do the following:
   - Select a Product Pack
   - Select a Platform
6. Click **Go**.
   The system displays a list of software for the selected product pack and platform.
7. Select the product that you want to download, and click **Continue**.
8. On the Download page, do the following:
   a) Click **Readme** for download instructions and information about the Media Pack.
   b) Click **Download** for the product download.
   c) Click **Download** for the License Document download.
9. On the Oracle Software Delivery Cloud tool bar, click **Sign Out**.

Next Steps

You must unzip all of the files associated with a specific product into the same directory. You must also keep the directories for different products separate from each other. The directory in which you unzip the product files will be the staging area from where you will install the software.

Installation and Start-Up

After you have completed the hardware installation procedures outlined in the the relevant *Hardware Installation Guide*, you are ready to establish a connection to your Oracle Enterprise Session Border Controller. Then you can load the software image you want to use and establish basic operating parameters.

Hardware Installation Process

Installing the Oracle Enterprise Session Border Controller hardware in a rack requires the following process.

1. Unpack the Oracle Enterprise Session Border Controller hardware.
2. Install the Oracle Enterprise Session Border Controller hardware into the rack.
3. Install the power supplies.
4. Install the fan modules.
5. Install the physical interface cards.
6. Cable the Oracle Enterprise Session Border Controller hardware.

Note: Complete installation procedures fully and note the safety warnings to prevent physical harm to yourself and damage to the Oracle Enterprise Session Border Controller hardware.

For more information, see the hardware documentation.

Connecting to Your Oracle Enterprise Session Border Controller

You can connect to your Oracle Enterprise Session Border Controller either through a direct console connection, or by creating a remote Telnet or SSH session. Both of these access methods provide you with the full range of configuration, monitoring, and management options.

Note: By default, Telnet and FTP connections to your Oracle Enterprise Session Border Controller are enabled.

Create a Console Connection

Using a serial connection, you can connect your laptop or PC directly to the Acme Packet hardware. If you use a laptop, you must take appropriate steps to ensure grounding.

One end of the cable plugs into your terminal, and the other end plugs into the RJ-45 Console port on the NIU (or management ports area on the Acme Packet 6300).

To make a console connection to your hardware:

1. Set the connection parameters for your terminal to the default boot settings:
   - Baud rate: 115,200 bits/second
   - Data bits: 8
   - Parity: No
   - Stop bit: 1
   - Flow control: None
2. Connect a serial cable to between your PC and the hardware's console port.
3. Apply power to the hardware.
4. Enter the appropriate password information when prompted to log into User mode of the ACLI.

   You can set the amount of time it takes for your console connection to time out by setting the **console-timeout** parameter in the system configuration. If your connection times out, the login sequence appears again and prompts you for your passwords. The default for this field is 0, which means that no time-out is being enforced.

Incoming Telnet Connections and Time-outs

You can Telnet to your Oracle Enterprise Session Border Controller. Using remote Telnet access, you can provision the Oracle Enterprise Session Border Controller remotely through the management IP interface.

The Oracle Enterprise Session Border Controller, when running on Acme Packet platforms can support up to 5 concurrent Telnet sessions. When running on other platform types, only 4 concurrent Telnet sessions are available. In both cases, only one Telnet session may be in configuration mode at a time.

Note: Telnet does not offer a secure method of sending passwords. Using Telnet, passwords are sent in clear text across the network.

To Telnet to your Oracle Enterprise Session Border Controller, you need to know the IP address of its administrative interface (wancom0/eth0). The wancom0/eth0 IP address of your Oracle Enterprise Session Border Controller is found by checking the **inet on ethernet** value in the boot parameters or visible from the front panel display.

You can manage incoming Telnet connections from the ACLI:

- To set a time-out due to inactivity, use the **telnet-timeout** parameter in the system configuration. You can set the number of seconds that elapse before the Telnet connection or SSH connection is terminated. The default for this field is 0, which means that no time-out is being enforced.
- To view the users who are currently logged into the system, use the ACLI **show users** command. You can see the ID, timestamp, connection source, and privilege level for active connections.

- From Superuser mode in the ACLI, you can terminate the connections of other users in order to free up connections. Use the **kill user** command with the corresponding connection ID.
- From Superuser mode in the ACLI, you can globally enable and disable Telnet connections:
  - Telnet service is enabled by default unless explicitly disabled as shipped.
  - To disable Telnet, type the **management disable telnet** command at the Superuser prompt and reboot your system. The Oracle Enterprise Session Border Controller then refuses any attempts at Telnet connections. If you want to restart Telnet service, type **management enable telnet**.
- If you reboot your Oracle Enterprise Session Border Controller from a Telnet session, you lose IP access and therefore your connection.

SSH Remote Connections

For increased security, you can connect to your Oracle Enterprise Session Border Controller using SSH. An SSH client is required for this type of connection.

The Oracle Enterprise Session Border Controller supports five concurrent SSH and/or SFTP sessions.

There are two ways to use SSH to connect to your Oracle Enterprise Session Border Controller. The first works the way a Telnet connection works, except that authentication takes place before the connection to the Oracle Enterprise Session Border Controller is made. The second requires that you set an additional password.

1. To initiate an SSH connection to the Oracle Enterprise Session Border Controller without specifying users and SSH user passwords:
   a) Open your SSH client (with an open source client, etc.).
   b) At the prompt in the SSH client, type the ssh command, a Space, the IPv4 address of your Oracle Enterprise Session Border Controller, and then press Enter.

      The SSH client prompts you for a password before connecting to the Oracle Enterprise Session Border Controller. Enter the Oracle Enterprise Session Border Controller's User mode password. After it is authenticated, an SSH session is initiated and you can continue with tasks in User mode or enable Superuser mode.

      Note: You can also create connections to the Oracle Enterprise Session Border Controller using additional username and password options.
2. To initiate an SSH connection to the Oracle Enterprise Session Border Controller with an SSH username and password:
   a) In the ACLI at the Superuser prompt, type the ssh-password and press Enter. Enter the name of the user you want to establish. Then enter a password for that user when prompted. Passwords do not appear on your screen.

      ```
      ACMEPACKET# ssh-password
      SSH username [saved]: MJones
      Enter new password: 95X-SD
      Enter new password again: 95X-SD
      ```

      After you configure ssh-password, the SSH login accepts the username and password you set, as well as the default SSH/SFTP usernames: User and admin.
   b) Configure your SSH client to connect to your Oracle Enterprise Session Border Controller's management IPv4 address using the username you just created. The standard version of this command would be:

      ```
      ssh -l MJones 10.0.1.57
      ```
   c) Enter the SSH password you set in the ACLI.

      ```
      MJones@10.0.2.54 password: 95X-SD
      ```
   d) Enter your User password to work in User mode on the Oracle Enterprise Session Border Controller. Enable Superuser mode and enter your password to work in Superuser mode.
   e) A Telnet session window opens and you can enter your password to use the ACLI.

System Boot

When your Oracle Enterprise Session Border Controller boots, the following information about the tasks and settings for the system appear in your terminal window.

- System boot parameters
- From what location the software image is being loaded: an external device or internal flash memory
- Requisite tasks that the system is starting
- Log information: established levels and where logs are being sent
- Any errors that might occur during the loading process

After the loading process is complete, the ACLI login prompt appears.

Oracle Enterprise Session Border Controller Boot Parameters

Boot parameters specify the information that your Oracle Enterprise Session Border Controller uses at boot time when it prepares to run applications. The Oracle Enterprise Session Border Controller's boot parameters:

- Allow you to set the IP address for the management interface (wancom0).
- Allow you to set a system prompt. The target name parameter also specifies the title name displayed in your web browser and SNMP device name parameters.
- Determine the software image to boot and from where the system boots that image.
- Sets up the username and password for network booting from an external FTP server.

In addition to providing details about the Oracle Enterprise Session Border Controller's boot parameters, this section explains how to view, edit, and implement them.

When displaying the boot parameters, your screen shows a help menu and the first boot parameter (boot device). Press Enter to continue down the list of boot parameters.

Boot Parameters

- boot device—The boot device for the Oracle Enterprise Session Border Controller should be eth0.
- file name—The file name for the Oracle Enterprise Session Border Controller should start with /boot/ (if local).

Sample Oracle Enterprise Session Border Controller Boot Parameters

The full set of Oracle Enterprise Session Border Controller boot parameters appears like the ones in this sample:

```
ACMEPACKET(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device             : eth0
processor number     : 0
host name     : acmepacket8
file name     : /boot/nnSC600.gz
inet on ethernet (e)     : 10.0.1.57:ffff0000
inet on backplane (b)     : 0.0.0.0
host inet (h)             : 10.0.1.5
gateway inet (g)          : 10.0.0.1
user (u)                  : user
ftp password (pw)         : password
flags (f)                 : 0x08
target name (tn)          : acmesystem
startup script (s)        : 0
other (o)                 :
NOTE: These changed parameters will not go into effect until reboot. Also, be
aware that some boot parameters may also be changed through the PHY and
Network Interface Configurations.
```

Notes on Boot Parameters

- The boot device in the bootparams for Acme Packet hardware is **eth0** .

- The standard path for image files is /boot.

Boot Parameter Changes

You can access and edit boot parameters by using either the ACLI or by interrupting the system boot process.

Note: Changes to boot parameters do not go into effect until you reboot the Oracle Enterprise Session Border Controller.

Oracle recommends that you use management port 0 (wancom0) as the boot interface, and that your management network is either:

- directly a part of your LAN for management port 0
- accessible through management port 0

Otherwise, your management messages may use an incorrect source address.

Change Boot Parameters from the ACLI

To access and change boot parameters from the ACLI:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type bootparam, and press Enter. The boot device parameters display.

```
ACMEPACKET(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device          : eth0
```

   To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

   You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one. You can clear the contents of a parameter by typing a period and then pressing Enter.

```
ACMEPACKET(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device     : eth0
processor number    : 0
host name    : goose
file name    : /boot/nnPCz100.gz /boot/nnPCz200.gz
```

   When you have scrolled through all of the boot parameters, the system prompt for the configure terminal branch displays.

```
ACMEPACKET(configure)#
```

4. Exit the configure terminal branch.
5. Reboot the Oracle Enterprise Session Border Controller for the changes to take effect.

   The ACLI reboot and reboot force commands initiate a reboot. With the reboot command, you must confirm that you want to reboot. With the reboot force command, you do not have make this confirmation.

```
ACMEPACKET# reboot force
```

   The Oracle Enterprise Session Border Controller completes the full booting sequence. If necessary, you can stop the auto-boot at countdown to fix any boot parameters.

   If you configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

   Note: If you configured the boot parameters incorrectly, the Oracle Enterprise Session Border Controller goes into a booting loop and displays an error message.

```
Error loading file: errno = 0x226.
Can't load boot file!!
```

Press the space bar to stop the loop. Correct the error in the boot parameter, and reboot the system.

Change Boot Parameters by Interrupting a Boot in Progress

To access and change boot parameters by interrupting a boot in progress:

1. When the Oracle Enterprise Session Border Controller is in the process of booting, you can press the space bar on your keyboard to interrupt when you see the following message appear:

   ```
   Press the space bar to stop auto-boot...
   ```

2. After you stop the booting process, you can enter the letter p to display the current parameters, the letter c to change the boot parameters or the @ (at-sign) to continue booting.

   ```
   [Acme Packet Boot]: c
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   boot device          : wancom0
   ```

   To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

   You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one.

   ```
   ACMEPACKET(configure)# bootparam
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   boot device     : wancom0
   processor number     : 0
   host name     : goose
   file name     : /code/nnPCz100.bz /code/nnPCz200.bz
   ```

4. After you have scrolled through the complete list of boot parameters, you return to the boot prompt. To reboot with your changes taking effect, type @ (the at-sign), and press Enter.

   ```
   [Acme Packet Boot]: @
   ```

   The Oracle Enterprise Session Border Controller completes the full booting sequence, unless there is an error in the boot parameters.

   If you have configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

   Note: If you have configured the boot parameters incorrectly, the Oracle Enterprise Session Border Controller goes into a booting loop and displays an error message.

   ```
   Error loading file: errno = 0x226.
   Can't load boot file!!
   ```

   Press the space bar to stop the loop. Correct the error, and reboot your system.

Boot Parameter Definitions

The following table defines each of the Oracle Enterprise Session Border Controller's boot parameters.

Boot Parameter	Description
boot device	Management interface name and port number of the device from which an image is downloaded (e.g., wancom0 or eth0) from an external device.
processor number	Processor number on the backplane.
host name	Name of the boot host used when booting from an external device.
file name	Name of the image file to be booted; can be entered with the filename path.

Boot Parameter	Description
	If you are booting from the flash memory, this filename must always match the filename that you designate when you FTP the image from the source to the Oracle Enterprise Session Border Controller. When booting from internal flash memory, this filename must start with /boot); for example, /boot/nnECZ710.bz.
inet on ethernet (e)	Internet address of the Oracle Enterprise Session Border Controller. This field can have an optional subnet mask in the form inet_adrs:subnet_mask. If DHCP is used to obtain the parameters, lease timing information may also be present. This information takes the form of lease_duration:lease_origin and is appended to the end of the field. In this parameter, the subnet mask ffff0000 = 255.255.0.0. When you use the ACLI acquire-config command, this is the IPv4 address of the Oracle Enterprise Session Border Controller from which you will copy a configuration.
inet on backplane (b)	Internet address of the backplane interface, eth0. This parameter can have an optional subnet mask and/or lease timing information, such as e (inet on ethernet) does.
host inet (h)	Internet address of the boot host used when booting from an external device.
gateway inet (g)	Internet address of the gateway to the boot host. Leave this parameter blank if the host is on the same network.
user (u)	FTP username on the boot host.
ftp password (pw)	FTP password for the FTP user on the boot host.
flags (f)	Codes that signal the Oracle Enterprise Session Border Controller from where to boot. Also signals the Oracle Enterprise Session Border Controller about which file to use in the booting process. This sequence always starts with 0x (these flags are hexadecimal). The most common codes are: 0x08: Means that the system looks at the filename defined in the boot configuration parameters to determine where to boot from and what file to use. If the file name parameter contains /tffsX/filename, then the system boots off the flash memory (see options below). If the file name parameter just contains a filename, then the Oracle Enterprise Session Border Controller boots off the external host defined and looks for the filename in the /tftpboot directory on that host. 0x80008: Used for source routing. If your requirements differ from what these flags allow, contact your Oracle customer support representative for further codes.
target name (tn)	Name of the Oracle Enterprise Session Border Controller as it appears in the system prompt. For example, ACMEPACKET> or ACMEPACKET#. You need to know the target name if you are setting up an HA node. This name is required to be unique among Oracle Enterprise Session Border Controllers in your network. This name can be 64 characters or less.
startup script (s)	For Oracle use only.

Boot Parameter	Description
other (o)	For Oracle use only.

Configurable Boot Loader Flags

You may configure the following boot flags in the boot loader:

- 0x04 - disables autoboot timeout (ap3820 and ap4500 only)
- 0x08 - extend autoboot countdown timer to 15 seconds
- 0x40 - use DHCP for wancom0 (VM Edition only)
- 0x80 - network boot using TFTP instead of FTP

Installation Wizard

The Oracle Enterprise Session Border Controller Wizard provides a wizard for initial system setup. The wizard installs the following elements.

- Management IP address & Gateway IP address
- Web GUI
- High Availability settings
- Net-Net Central (NNC) access setting

This section provides information about the Installation Wizard and details a step-by-step procedure you can follow while using the Wizard.

About the Installation Wizard

The Installation Wizard allows you to simply and easily install basic system elements, including the Web GUI, by answering a series of questions that elicit basic configuration responses. All questions provide an intelligent default value that can be accepted by the user to enable reliable and consistent system connectivity.

After completing the Installation Wizard, you can perform detailed, Oracle Enterprise Session Border Controller configuration using the Web GUI.

The Web GUI provides a configuration tab that allows you to configure the Oracle Enterprise Session Border Controller. This tab provides two methods by which you can configure:

- Basic Mode - Drag and drop elements onto a workspace to configure the Oracle Enterprise Session Border Controller (recommended for most users)
- Expert Mode - Configure the Oracle Enterprise Session Border Controller using a parameter configuration tree. (recommended for complex configurations which are unique).

During the Installation Wizard setup, the Basic Mode is installed automatically. However, you can choose to enable or disable the Expert Mode if required. If you disable the Expert Mode, only Basic Mode is accessible in the Web GUI. If you enable Expert Mode, and you make changes to the configuration, you can switch between Expert Mode and Basic Mode but you must save and activate the configuration before you can switch modes.

After the Installation Wizard is complete, if you decide you want to enable Expert Mode, you must re-run the Installation Wizard from the ACLI and select Expert Mode during the installation.

Note: If required, you can also setup High Availability (HA) using this Installation Wizard. To setup HA, see *Setting Up High Availability (HA) Mode*.

You can setup the Web Server/Web GUI manually, if required, using the procedures provided in Appendix D *Manual Web Server Configuration (1263)*.

Running Setup

Use the following procedure to run the Installation Wizard to configure the Web Server and setup your Web GUI.

To run the Installation Wizard:

1. In Superuser mode, enter run setup and press Enter.

```
ACMEPACKET# run setup
```

Note: The Installation Wizard can also be invoked by the run setup quiet command which enables a less verbose presentation.

The following displays.

```
============================================================
------------------------------------------------------------
Thank you for purchasing the Oracle SBC. The following
short wizard will guide you through the initial set-up.
------------------------------------------------------------
'?' = Help; '.' = Clear; 'q' = Exit

CONFIGURATION

WARNING: Proceeding with wizard will result in existing configuration being
erased.
  Erase config and proceed (yes/no) [no]                :
============================================================
```

You can use:

- '?' key to obtain query-specific help
- '.' key to clear the field of the current setting.
- 'q' key to exit the wizard at any location in the setup process. Initiating the q displays the following prompt:

```
Discarding changes and quitting wizard. Are you sure? [y/n]?:
```

Enter y to discard any changes and quit the installation wizard. The root prompt displays.

A "Warning" displays stating that using the wizard can overwrite (erase) the existing running configuration. If you want to back out of the setup process and not overwrite the current running configuration, you can enter q or enter No at the Erase config and proceed prompt. The root prompt displays.

2. Enter yes at the prompt to continue the setup process, and press Enter.

```
Erase config and proceed (yes/no) [no]     :yes
```

The following displays.

```
============================================================
Configuration will be backed up as
bkup_setup_wizard_<MMM>_<DD>_<YY>_<HH>_<MM>_<SSS>.gz
'-' = Previous; '?' = Help; '.' = Clear; 'q' = Exit
GUI ACCESS
If you want to allow GUI to access this SBC, enable this setting
Enable Web GUI (yes/no) [yes]     :
============================================================
```

You can use:

'-' key to navigate to the previous step in the setup process, if required, and change your response.

3. Enter yes and press Enter. Or enter no to disable the Web GUI and press Enter.

```
Enable Web GUI (yes/no) [yes]     :yes
```

The following displays.

```
============================================================
WEB GUI MODE
Choose which mode to enable for the web GUI
  Web GUI Mode
     1 - basic
     2 - expert
```

```
    Enter choice [1 - basic]     :
=========================================================
```

Enable Basic Mode and or Expert Mode

1. Enter 1 to enable Basic Mode and Expert Mode or 2 to enable Expert Mode only. Press Enter.

```
Enter choice [1 - basic]     :1
```

Note: If you enable both Basic and Expert Modes (selection 1), after logging into the Web GUI, the Basic Mode screen displays by default. You can switch to Expert Mode if you do not make any changes in Basic Mode, using the Switch to Expert button. If you enable Expert Mode only (selection 2), no button displays to switch to Basic Mode.

The following displays.

```
=========================================================
HIGH AVAILABILITY
This SBC may be a standalone or part of a highly available redundant pair.
  SBC mode
     1 - standalone
     2 - high availability
    Enter choice [1 - standalone]     :
=========================================================
```

2. Enter 1 for a standalone server or enter 2 to configure an HA device. Press Enter and go to Step 4 in *Setting Up High Availability (HA) Mode* to complete the installation.

Note: It is highly recommended that you configure high availability (HA) using this Installation Wizard.

```
Enter choice [1 - standalone]     :1
```

The following displays.

```
=========================================================
SBC SETTINGS
Unique target name of this SBC [<SBC name>]     :
=========================================================
```

3. Enter a unique target name for the NN-ESD (or keep the default in [ ]), and press Enter.

```
Unique target name of this SBC [<SBC name>]     :
```

The following displays.

```
=========================================================
IP address on management interface [<SBC IP address>] :
=========================================================
```

4. Enter the IP address to be used for accessing the Web GUI (or keep the default in [ ]), and press Enter.

```
IP address on management interface [<SBC IP Address>]     : 164.30.85.51
```

The following displays.

```
=========================================================
Subnet mask [<subnet mask>] :
=========================================================
```

5. Enter the subnet mask of the Oracle Enterprise Session Border Controller (or keep the default in [ ]), and press Enter.

```
Subnet mask [<subnet mask>]     : 255.255.0.0
```

The following displays.

```
=========================================================
Gateway IP address [<Gateway IP address>] :
=========================================================
```

6. Enter the gateway IP address (or keep the default in [ ]), and press Enter.

```
Gateway IP address [<Gateway IP address>]     :164.30.0.1
```

The following displays.

```
=========================================================
-- Summary view -----------------------------------
GUI ACCESS
  1: Enable Web GUI (yes/no)                           : yes
WEB GUI MODE
 2 : Web GUI Mode                                      : basic
HIGH AVAILABILITY
 3 : SBC mode                                          : standalone
 4 : SBC role                                          : N/A
SBC SETTINGS
 7 : Unique target name of this SBC                    : lise_primary
 8 : IP address on management interface                : 164.30.85.51
 9 : Subnet mask                                       : 255.255.0.0
 10: Gateway IP address                                : 164.30.0.1
AUTOMATIC CONFIGURATION
 11: Acquire config from the Primary (yes/no)          : N/A
PEER CONFIGURATION
 13: Peer target name                                  : N/A
Enter 1 - 16 to modify, 'd' to display summary, 's' to save, 'q' to exit.
[s]:
=========================================================
```

Note: Only the fields that were configured display in the summary view.

7. Enter s to save the configuration. Or select an item number from the summary view to modify the value for that item. Or enter q to exit the installation wizard without saving the setup.

```
Enter 1 - 16 to modify, 'd' to display summary, 's' to save, 'q' to exit.
[s]    :s
```

The following displays.

```
=========================================================
Saving changes and quitting wizard. Are you sure?  [y/n]?    :
=========================================================
```

8. Enter y to verify you want to save the configuration and press Enter.

```
Saving changes and quitting wizard. Are you sure?  [y/n]?    :y
```

The following displays.

```
=========================================================
Running configuration is backed up as
'bkup_setup_wizard_Apr_18_13_04_57_970.gz'
*********************************************************
Deleting configuration
Erase-Cache received, processing.
waiting 1200 for request to finish
Request to 'ERASE-CACHE' has Finished,
Erase-Cache: Completed
Request to 'RESTORE-CONFIG' has Finished,
Restore Backup Completed Successfully
checking configuration
Save-Config received, processing.
waiting for request to finish
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.
---------------------------------------------------
Verification successful! No errors nor warnings in the configuration
Activate-Config received, processing.
```

```
waiting for request to finish
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
-- Saved configuration. ---------------------------
GUI ACCESS
  1: Enable Web GUI (yes/no)                          : yes
WEB GUI MODE
 2 : Web GUI Mode                                     : basic
HIGH AVAILABILITY
 3 : SBC mode                                         : standalone
 4 : SBC role                                         : N/A
SBC SETTINGS
 7 : Unique target name of this SBC                   : jane_primary
 8 : IP address on management interface               : 164.30.85.51
 9 : Subnet mask                                      : 255.255.0.0
 10: Gateway IP address                               : 164.30.0.1
AUTOMATIC CONFIGURATION
 11: Acquire config from the Primary (yes/no)         : N/A
PEER CONFIGURATION
 13: Peer target name                                 : N/A
You may access the GUI via http://164.30.85.51:80/ after reboot.
============================================================
```

You have completed the Installation Wizard.

For more information about using the Web GUI, see the Net-Net Enterprise Session Director Web GUI User Guide.

You can configure the Web Server for using Transport Layer Security (TLS). You can also manage the Web Server using specific commands. For more information about configuring TLS on the Web Server, and for information about Web Server management commands, see Chapter 22, *Web Server TLS Configuration and Management Commands (1203)*.

Setting Up High Availability (HA) Mode

Use the following procedure to perform an HA configuration on a primary and secondary Oracle Enterprise Session Border Controller. You must have an HA license to configure HA.

Note: For HA environments, running setup on the secondary system is also required to set the wancom0 address and secondary targetname. The targetname must match the same secondary targetname specified on the primary system.

To configure the primary High Availability (HA):

1. In Superuser mode, enter run setup and press Enter.

```
ACMEPACKET# run setup
```

Note: The Installation Wizard can also be invoked by the run setup quiet command which enables a less verbose presentation.

The following displays.

```
============================================================
------------------------------------------------------------
Thank you for purchasing the Oracle  SBC. The following
short wizard will guide you through the initial set-up.
------------------------------------------------------------
'?' = Help; '.' = Clear; 'q' = Exit
CONFIGURATION
WARNING: Proceeding with wizard will result in existing configuration being
erased.
  Erase config and proceed (yes/no) [no]               :
============================================================
```

You can use:

- '?' key to obtain query-specific help
- '.' key to clear the field of the current setting.
- 'q' key to exit the wizard at any location in the setup process. Initiating the q displays the following prompt:

```
Discarding changes and quitting wizard. Are you sure? [y/n]?:
```

Enter y to discard any changes and quit the installation wizard. The root prompt displays.

A "Warning" displays stating that using the wizard can overwrite (erase) the existing running configuration. If you want to back out of the setup process and not overwrite the current running configuration, you can enter q or enter No at the Erase config and proceed prompt. The root prompt displays.

2. Enter yes at the prompt to continue the setup process, and press Enter.

```
Erase config and proceed (yes/no) [no]     :yes
```

The following displays.

```
=========================================================
Configuration will be backed up as
bkup_setup_wizard_<MMM>_<DD>_<YY>_<HH>_<MM>_<SSS>.gz
'-' = Previous; '?' = Help; '.' = Clear; 'q' = Exit
GUI ACCESS
If you want to allow GUI to access this SBC, enable this setting
Enable Web GUI (yes/no) [yes]     :
=========================================================
```

You can use:

- '-' key to navigate to the previous step in the setup process, if required, and change your response.

3. Enter yes or no to enable or disable the Web GUI and press Enter.

```
Enable Web GUI (yes/no) [yes]     :yes
```

The following displays.

```
=========================================================
WEB GUI MODE
Choose which mode to enable for the web GUI
  Web GUI Mode
     1 - basic
     2 - expert
    Enter choice [1 - basic]     :
=========================================================
```

4. Enter 1 to enable Basic Mode and Expert Mode, or enter 2 to enable Expert Mode only. Press Enter.

```
Enter choice [1 - basic]     :1
```

If you enable both Basic and Expert Modes (selection 1), after logging into the Web GUI, the Basic Mode screen displays by default. You can switch to Expert Mode if you do not make any changes in Basic Mode, using the Switch to Expert button. If you enable Expert Mode only (selection 2), no button displays to switch to Basic Mode.

The following displays.

```
=========================================================
HIGH AVAILABILITY
This SBC may be a standalone or part of a highly available redundant pair.
  SBC mode
     1 - standalone
     2 - high availability
    Enter choice [1 - standalone]     :
=========================================================
```

5. Enter 2 to configure an HA device, press Enter.

It is highly recommended that you configure high availability (HA) using this Installation Wizard.

```
Enter choice [1 - standalone]     :2
```

The following displays.

```
=============================================================
If this SBC is the primary, enter the configuration. If it is the
secondary, you can import settings from the primary.
SBC role
    1. primary
    2. secondary
Enter choice [1-primary]        :
=============================================================
```

6. Enter 2 and press Enter.

```
Enter choice [1-primary]        :2
```

The following displays:

```
=============================================================
SBC SETTINGS
Unique target name of this SBC [<NN-ESD name>]     :
=============================================================
```

7. Enter a unique target name for the NN-ESD (or keep the default in [ ]), and press Enter.

```
Unique target name of this SBC [NNESD1]     :NNESD2
```

The following displays.

```
=============================================================
SBC SETTINGS
IP address on management interface [<SBC IP address>] :
=============================================================
```

Note: The Installation Wizard provides default IP addresses for the HA primary and secondary Oracle Enterprise Session Border Controllers (Redundancy interface address and Peer IP address). These default addresses are link-local addresses as specified in RFC 3927, *Dynamic Configuration of IPv4 Link-local addresses*.

8. Enter the IP address on the management interface of the secondary Web Server (or keep the default in [ ]), and press Enter.

```
IP address on management interface [<SBC IP address>]     : 164.30.85.52
```

The following displays.

```
Subnet mask [255.255.0.0] :
=============================================================
```

9. Enter the subnet mask of the secondary Web Server (or keep the default in [ ]), and press Enter.

```
Subnet mask [255.255.0.0]     : 255.255.0.0
```

The following displays.

```
=============================================================
Gateway IP address [164.30.0.1]     :
=============================================================
```

10. Enter the gateway IP address of the secondary Web Server (or keep the default in [ ]), and press Enter.

```
Gateway IP address [<SBC gateay address>]     : 164.30.0.1
```

The following displays.

```
=============================================================
AUTOMATIC CONFIGURATION
Acquire config from the Primary (yes/no) [yes]    :
=============================================================
```

11. Enter y for the secondary Web Server to acquire the configuration from the primary Web Server during failover, and press Enter.

```
Acquire config from the Primary (yes/no) [yes]    y
```

The following displays.

```
============================================================
-- Summary view ------------------------------------
GUI ACCESS
  1: Enable Web GUI (yes/no)     : yes
WEB GUI MODE
 2 : Web GUI Mode     : basic
HIGH AVAILABILITY
 3 : SBC mode     : high availability
 4 : SBC role     : secondary
SBC SETTINGS
 7 : Unique target name of this SBC     : NNESD2
 8 : IP address on management interface     : 164.30.85.52
 9 : Subnet mask     : 255.255.0.0
 10: Gateway IP address     : 164.30.0.1
AUTOMATIC CONFIGURATION
 11: Acquire config from the Primary (yes/no)     : yes
PEER CONFIGURATION
 13: Peer target name     : N/A
Enter 1 - 16 to modify, 'd' to display summary, 's' to save, 'q' to exit.
[s]:
============================================================
```

12. Enter s to save the configuration and press Enter. Or Select an item number from the summary view to modify the value for that item. Or Enter q to exit the installation wizard.

```
Enter 1 - 16 to modify, 'd' to display summary, 's' to save, 'q' to exit.
[s]    :s
```

The following displays.

```
============================================================
Saving changes and quitting wizard. Are you sure?  [y/n]?    :
============================================================
```

Enter y to verify you want to save the configuration and press Enter.

The following displays.

```
============================================================
Running configuration is backed up as
'bkup_setup_wizard_Mar__7_13_58_26_545.gz'
**************************************************
Deleting configuration
Erase-Cache received, processing.
waiting 1200 for request to finish
Request to 'ERASE-CACHE' has Finished,
Erase-Cache: Completed
Request to 'RESTORE-CONFIG' has Finished,
Restore Backup Completed Successfully
checking configuration
Save-Config received, processing.
waiting for request to finish
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
Activate-Config received, processing.
waiting for request to finish
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
-- Saved configuration. ---------------------------
GUI ACCESS
```

```
    1: Enable Web GUI (yes/no)     : yes
WEB GUI MODE
 2 : Web GUI Mode     : basic
HIGH AVAILABILITY
 3 : SBC mode     : high availability
 4 : SBC role     : secondary
SBC SETTINGS
 7 : Unique target name of this SBC     : NNESD2
 8 : IP address on management interface     : 164.30.85.52
 9 : Subnet mask     : 255.255.0.0
 10: Gateway IP address     : 164.30.0.1
AUTOMATIC CONFIGURATION
 11: Acquire config from the Primary (yes/no)     : yes
PEER CONFIGURATION
 13: Peer target name     : N/A
You may access the GUI via http://164.30.85.52:80/ after reboot.
=========================================================
```

You have completed the Installation Wizard. To complete your HA setup, run the Installation Wizard on your secondary system as well.

For more information about using the Web GUI, see the *Net-Net Enterprise Session Director-Web GUI User Guide*.

You can configure the Web Server for using Transport Layer Security (TLS). You can also manage the Web Server using specific commands. For more information about configuring TLS on the Web Server, and for information about Web Server management commands, see Chapter 22, *Web Server TLS Configuration and Management Commands (1203)*.

Setting Up System Basics

Before configuring and deploying the Oracle Enterprise Session Border Controller, you might want to establish some basic attributes such as a system prompt, new User and Superuser passwords, and NTP synchronization.

New System Prompt

The ACLI system prompt is set in the boot parameters. To change it, access the boot parameters and change the target name value to make it meaningful within your network. The target name may be up to 38 characters. A value that identifies the system in some way is often helpful.

NTP Synchronization

This section provides information about how to set and monitor NTP on your Oracle Enterprise Session Border Controller.

When an NTP server is unreachable or when NTP service goes down, the Oracle Enterprise Session Border Controller generates traps for those conditions. Likewise, the Oracle Enterprise Session Border Controller clears those traps when the conditions have been rectified. The Oracle Enterprise Session Border Controller considers a configured NTP server to be unreachable when its reach number (whether or not the NTP server could be reached at the last polling interval; successful completion augments the number) is 0. You can see this value for a server when you use the ACLI show ntp server command.

- The traps for when a server is unreachable and then again reachable are: apSysMgmtNTPServerUnreachableTrap and apSysMgmtNTPServerUnreachableClearTrap
- The traps for when NTP service goes down and then again returns are: apSysMgmtNTPServiceDownTrap and apSysMgmtNTPServiceDownClearTrap

Setting NTP Synchronization

When the Oracle Enterprise Session Border Controller requires time-critical processing, you can set NTP for time synchronization. Setting NTP synchronizes both the hardware and the software clocks with the reference time from an

NTP server that you specify. NTP is most useful for synchronizing multiple devices located on one network, or across many networks, to a reference time standard.

To guard against NTP server failure, NTP is restarted periodically to support the dynamic recovery of an NTP server.

You can only set NTP synchronization from the ACLI, but you can view it from the EMS. NTP is RTC-supported as of Net-Net OS Release C5.1.

Note that ntp-sync works only by way of the management interface and only on wancom0. Do not configure ntp-sync by way of the media interface or any other port.

To set NTP synchronization:

1. In the ACLI's configure terminal section, type ntp-sync and press Enter to access the NTP configuration. For example:

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# ntp-sync
ACMEPACKET(ntp-config)#
```

2. To add an NTP server, type add-server, a Space, the IPv4 address of the server, and then press Enter.

   For example, this entry adds the NTP server at the Massachusetts Institute of Technology in Cambridge, MA:

```
ACMEPACKET(ntp-config)# add-server 18.26.4.105
```

3. To delete an NTP server, type delete-server and the IPv4 address of the server you want to delete, and then press Enter.

```
ACMEPACKET(ntp-config)# del-server 18.26.4.105
```

Authenticated NTP

The Oracle Enterprise Session Border Controller can authenticate NTP server requests using MD5. The configured MD5 keys are encrypted and obscured in the ACLI. You configure an authenticated NTP server with its IP address, authentication key, and the key ID. Corresponding key and key IDs are provided by the NTP server administrator.

To configure an authenticated NTP server:

1. Access the ntp-config configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# ntp-sync
ACMEPACKET(ntp-config)#
```

2. Type select.

```
ACMEPACKET(ntp-config)# select
```

3. Access the auth-servers configuration element

```
ACMEPACKET(ntp-config)# auth-servers
ACMEPACKET(auth-servers)#
```

4. ip-address — Enter the IP address of the NTP server that supports authentication.
5. key-id — Enter the key ID of the key you enter in the next step. This value's range is 1 - 999999999.
6. key — Enter the key used to secure the NTP requests. The key is a string 1 - 31 characters in length.
7. Type done to save your work.
8. Type exit to return to the previous configuration level.
9. Type done to save the parent configuration element.

Monitoring NTP from the ACLI

NTP server information that you can view with the new show ntp server command tell you about the quality of the time being used in terms of offset and delays measurements. You can also see the maximum error bounds.

When you use this command, information for all configured servers is displayed. Data appears in columns that are defined in the table below:

Display Column	Definition
server	Lists the NTP servers configured on the Oracle Enterprise Session Border Controller by IP address. Entries are accompanied by characters: Plus sign (+)—Symmetric active server Dash (-)—Symmetric passive server Equal sign (=)—Remote server being polled in client mode Caret (^)—Server is broadcasting to this address Tilde (~)—Remote peer is sending broadcast to * Asterisk (*)—The peer to which the server is synchronizing
st	Stratum level—Calculated from the number of computers in the NTP hierarchy to the time reference. The time reference has a fixed value of 0, and all subsequent computers in the hierarchy are n+1.
poll	Maximum interval between successive polling messages sent to the remote host, measured in seconds.
reach	Measurement of successful queries to this server; the value is an 8-bit shift register. A new server starts at 0, and its reach augments for every successful query by shifting one in from the right: 0, 1, 3, 7, 17, 37, 77, 177, 377. A value of 377 means that there have been eight successful queries.
delay	Amount of time a reply packet takes to return to the server (in milliseconds) in response.
offset	Time difference (in milliseconds) between the client's clock and the server's.
disp	Difference between two offset samples; error-bound estimate for measuring service quality.

View Statistics

To view statistics for NTP servers:

At the command line, type show ntp server and press Enter.

```
ACMEPACKET# show ntp server
NTP Status                                    FRI APR 11:09:50 UTC 2007
 server                  st  poll  reach   delay    offset     disp
----------------------- --  ----  ------  -------  --------   ---------
*64.46.24.66              3    64     377  0.00018  0.000329   0.00255
=61.26.45.88              3    64     377  0.00017  0.002122   0.00342
```

You can the see the status of NTP on your system by using the show ntp status command. Depending on the status of NTP on your system, one of the following messages will appear:

- NTP not configured
- NTP Daemon synchronized to server at [the IP address of the specific server]
- NTP synchronization in process
- NTP down, all configured servers are unreachable

View Status

To view the status of NTP on your Oracle Enterprise Session Border Controller:

At the command line, type show ntp status and press Enter.

```
ACMEPACKET# show ntp status
```

Using the Oracle Enterprise Session Border Controller Image

The Oracle Enterprise Session Border Controller arrives with the most recent, manufacturing-approved run-time image installed on the flash memory. If you want to use this image, you can install the Oracle Enterprise Session Border Controller as specified in the *Acme Packet Hardware Installation Guide* , establish a connection to the Oracle Enterprise Session Border Controller, and begin to configure it. On boot up, the system displays information about certain configurations not being present. You can dismiss these displays and begin configuring the Oracle Enterprise Session Border Controller.

If you want to use an image other than the one installed on the Oracle Enterprise Session Border Controller when it arrives, you can use the information in this section to obtain and install the image.

Obtaining a New Image

You can download a software image onto the Oracle Enterprise Session Border Controller platform from the following sources.

- Obtain an image from the FTP site and directory where you or your Oracle customer support representative placed the image. For example, this may be a special server that you use expressly for images and backups.
- Obtain an image from your Oracle customer support representative, who will transfer it to your system.

Regardless of the source, you can use FTP or SFTP to copy the image from the source to the Oracle Enterprise Session Border Controller.

Copy an Image to the Oracle Enterprise Session Border Controller using FTP

The /boot directory on the Oracle Enterprise Session Border Controller has 32mb available, and operating system files of approximately 9mb each. Oracle recommends storing no more than two images at a time in this location. One of these should be the latest version. The /boot directory is used for the on-board system flash memory. If you do not put the image in this directory, the Oracle Enterprise Session Border Controller will not find it.

Note: You can also use SFTP.

To copy an image on your Oracle Enterprise Session Border Controller using FTP:

1. Go to the directory where the image is located.
2. Check the IP address of the Oracle Enterprise Session Border Controller's management port (wancom0). (You might think of this as a management address since it is used in the management of your Oracle Enterprise Session Border Controller.)
3. Create the connection to the Oracle Enterprise Session Border Controller. In your terminal window, type ftp and the IPv4 address of the Oracle Enterprise Session Border Controller management port (wancom0), and press Enter. Once a connection has been made, the system displays a confirmation note followed by the FTP prompt.
4. Enter your FTP username and FTP password information. The username is always user, and the password is the same as the one you use for the User mode login.
5. Go to the directory where you want to put the image.
6. From the FTP prompt, do the following:

   - Change the directory to /boot.

     ```
     ftp> cd /boot
     ```

   - Invoke binary mode.

     ```
     ftp> binary
     ```

     Note: Be sure to use binary transfer mode. If you do not, all transfers will be corrupted.

   - At the FTP prompt, enter the put command, a Space, the name of the image file, and press Enter.

```
ftp> put [file name]
```

The system displays confirmation that the connection is opening and that transfer is taking place.

- After the file transfer is complete, you can quit.

```
ftp> quit
```

7. Boot the Oracle Enterprise Session Border Controller using the image you just transferred.

   In the ACLI, change any boot configuration parameters that need to be changed. It is especially important to change the filename boot parameter to the filename you used during the FTP process. Otherwise, your system will not boot properly.

   Alternatively, from the console you can reboot to access the boot prompt and then configure boot parameters from there.
8. In the ACLI, execute the save-config command in order to save your changes.
9. Reboot the Oracle Enterprise Session Border Controller.
10. The Oracle Enterprise Session Border Controller runs through its loading processes and returns you to the ACLI logon prompt.

System Image Filename

The system image filename is a name you set for the image. This is also the filename the boot parameters uses when booting your system. This filename must match the filename specified in the boot parameters. When you use it in the boot parameters, it should always start with /boot to signify that the Oracle Enterprise Session Border Controller is booting from the /boot directory.

If the filename set in the boot parameters does not point to the image you want sent to the Oracle Enterprise Session Border Controller via FTP, then you could not only fail to load the appropriate image, but you could also load an image from a different directory or one that is obsolete for your purposes. This results in a boot loop condition that you can fix by stopping the countdown, entering the appropriate filename, and rebooting the Oracle Enterprise Session Border Controller.

Booting an Image on Your Oracle Enterprise Session Border Controller

You can either boot your Oracle Enterprise Session Border Controller from the system's local storage or from an external device. Both locations can store images from which the system can boot. This section describes both booting methods.

For boot parameters to go into effect, you must reboot your Oracle Enterprise Session Border Controller. Since a reboot stops all call processing, Oracle recommends performing tasks that call for a reboot during off-peak hours. If your Oracle Enterprise Session Border Controllers are set up in an HA node, you can perform these tasks on the standby system first.

Booting from Flash Memory

Once you have installed an image, you can boot your Oracle Enterprise Session Border Controller from its flash memory. With the exception of testing an image before you install it on the flash memory, this is generally the method you use for booting.

To boot from your Oracle Enterprise Session Border Controller flash memory:

1. Confirm that the boot parameters are set up correctly, and make any necessary changes.

   You can check the boot configuration parameters by accessing the bootparam command from the configure terminal menu.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET# bootparam
   ```
2. Change any boot configuration parameters that you need to change. It is especially important to change the file name boot configuration parameter. The file name parameter needs to use the /boot value so that the Oracle Enterprise Session Border Controller boots from the flash.
3. Reboot your Oracle Enterprise Session Border Controller.

4. You are be returned to the ACLI login prompt. To continue with system operations, enter the required password information.

Booting from an External Device

Booting from an external device means that your Oracle Enterprise Session Border Controller connects to a server to retrieve the boot image at boot time. Rather than using an image stored on your system's flash memory, it downloads the image from the external device each time it reboots.

When you are testing a new image before putting it on your Oracle Enterprise Session Border Controller, you might want to boot from an external device. Ordinarily, you would not want to boot an image on your Oracle Enterprise Session Border Controller this way.

To boot an image from an external device:

1. Confirm that the Oracle Enterprise Session Border Controller is cabled to the network from which you are booting. This is port 0 on the rear panel of the Oracle Enterprise Session Border Controller chassis (wancom0). The image is loaded from the source using FTP.
2. Log into the system you want to mount.
3. On the Oracle Enterprise Session Border Controller, configure the information for the boot parameters and confirm the following:
   - boot device—device to which you will FTP

     This parameter value must contain the name of the applicable management interface, and then the number of the appropriate 10/100 port. Usually, this value is wancom0.
   - file name—name on the host of the file containing the image

     The image file must exist in the home directory of the user on the image source.
   - host inet—IPv4 address of the device off of which you are booting
   - gateway inet—IPv4 address of the gateway to use if the device from which you are booting is not on the same network as your Oracle Enterprise Session Border Controller
   - user—username for the FTP account on the boot host
   - password—password for the FTP account on the boot host
4. Reboot your Oracle Enterprise Session Border Controller.
5. You are returned to the ACLI login prompt. To continue with system operations, enter the required password information.

Customizing Your ACLI Settings

This section describes several ways you can customize the way you log into the ACLI and the way the ACLI displays information. Where applicable, these descriptions also contain instructions for configuration.

Disabling the Second Login Prompt

With this feature enabled, the Oracle logs you in as a Superuser (i.e., in administrative mode) regardless of your configured privilege level for either a Telnet or an SSH session. However, if you log via SSH, you still need to enter the password for local or RADIUS authentication.

Disabling the Second Login Prompt Configuration

You disable the second login prompt in the authentication configuration.

To disable the second login prompt:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type authentication and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(security)# authentication
ACMEPACKET(authentication)#
```

4. login-as-admin—Set this parameter to enabled if you want users to be logged automatically in Superuser (administrative) mode. The default for this parameter is disabled.
5. Save and activate your configuration.

Persistent ACLI more Parameter

To make using the ACLI easier, the Oracle Enterprise Session Border Controller provides a paging feature controlled through the ACLI cli more command (which you can set to enabled or disabled). Disabled by default, this feature allows you to control how the Oracle Enterprise Session Border Controller displays information on your screen during a console, Telnet, or SSH session. This command sets the paging feature on a per session basis.

Customers who want to set the paging feature so that settings persist across sessions with the Oracle Enterprise Session Border Controller can set a configuration parameter that controls the paging feature. Enabling this parameter lets you set your preferences once rather than having to reset them each time you initiate a new session with the Oracle Enterprise Session Border Controller.

Persistent ACLI more Parameter Configuration

To set the persistent behavior of the ACLI more feature across sessions:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system-config and press Enter.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

   If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI select command) before making your changes.

4. cli-more—Set this parameter to enabled if you want the ACLI more paging feature to work persistently across console, Telnet, or SSH sessions with the Oracle Enterprise Session Border Controller. If you want to continue to set this feature on a per session basis, leave this parameter set to disabled (default).
5. Save and activate your configuration.

Customized Login Banner

A text file can be put on the Oracle Enterprise Session Border Controller to be used as a banner to be printed before each login. The file must be called /code/banners/banner.txt. The contents of this file will be printed before each User Access Verification login sequence. The limits are that no more than 79 characters per line and no more than 20 lines from the banner.txt file will be printed.

The banner.txt file used for the ACLI customized login banner has to be saved in the /code/banners directory. If that directory does not already exist on your system, you do not have to create the directory prior to placing the banner file because the Oracle Enterprise Session Border Controller will create it upon boot if it does not exist.

ACLI Expected Behavior

The following table describes the expected behavior for some of the ACLI commands.

Description of Behavior
Entering options in the ACLI options parameters. Where there are values to specify with an option, you must enter the option name and the values within parentheses. This example shows the correct syntax: ACMEPACKET# options (Methods="INVITE,ACK,PRACK,CANCEL,BYE,INFO,UPDATE,OPTIONS",max-udp-length=0) The following is incorrect syntax: ACMEPACKET# options Methods="INVITE,ACK,PRACK,CANCEL,BYE,INFO,UPDATE,OPTIONS",max-udp-length=0
The ACLI does not support multiple parameters that accept empty strings entered with either quotation marks ( ) or parentheses (()).
The edit command does not display as part of the ACLI menu. You can select a configuration to perform edits using the ACLI select command. You can use the no command to delete configuration.
When you want to use spaces as part of a value you enter, the entire entry must be enclosed in quotation marks (“”). Example: ACMEPACKET# community-name Acme Packet
All available parameters for a configuration object are displayed, even when they might be rendered redundant by the presence of another configuration setting. For example, in the static flow configuration, setting the alg-type to none means you don’t need to set several other parameters in for that static flow. Regardless, that static flow configuration displays those parameters.
The session recording servers sub-element in the session recording group configuration supporting adding servers using the plus sign (+) and removing them with the minus sign (-).

All ACLI Help is displayed consistently. Example of new, consistent Help:
`trans-protocol-match`
 	
`<enumeration> transport protocol`
`Default: ALL`
`<TCP,`
`UDP,`
`ICMP,`
`SCTP,`
`IPV6-ICMP,`
`ALL>`
 	
Example of old, inconsistent Help:
`trans-protocol-match`
 	
`<enumeration> transport protocol (default: all)`
`<tcp, udp, icmp, sctp, ipv6-icmp, all>`

ACLI parameters appear in ACLI displayed output even when left empty (have no values configured).
All password values across the ACLI display as a series of 8 asterisks (*).
If the value you want to configure has spaces, you must enclose the enter value in quotation marks () or parentheses (()) for it to be accepted. When configuring values for dates and times, you must use the proper entry format. Entries for date and time entry display both the date and time and adheres to the yyyy-mm-dd-hh:mm:ss.zzz format where y=year, m=month, d=day, h=hours, m=minutes, s=seconds, and z=milliseconds. When an attribute is not set, the ACLI displays it as empty in any output.

Description of Behavior
If there are no sub-elements set up for a configuration that has them, the ACLI displays nothing (for the unconfigured sub-elements).
The value you set in one parameter has no impact on the values you set for any other parameter. This behavior allows you to set parameters in any order you want. The ACLI checks valid values when you execute the done command. For example, it used to be that if the action parameter for header manipulation rule was set to reject, the new value parameter could not be set to any value containing a colon (:).
For the content type parameter in the sip-mim-rules configuration, the value was previously converted to type[^] when you carried out the done command. Now the parameter is simply type. The old and new values are equivalent, and there is no impact to functionality.

5

Software Upgrades

Introduction

This chapter provides information about how to upgrade your Oracle Enterprise Session Border Controller software image.

Be sure to read the *Release Notes* for any hardware and software requirements to upgrade to this release.

Notes on Boot Parameters

- The boot device in the bootparams for Acme Packet hardware is **eth0** .
- The standard path for image files is /boot.

Password Secure Mode

Note that all Oracle Enterprise Session Border Controllers have password secure mode enabled—meaning that you must accurately track your password information. To learn more about password secure mode, refer to this guide's Data Storage Security section.

Upgrading Software Images

This document explains how to upgrade software images on your Oracle Enterprise Session Border Controller.

Upgrade Checklist

Before upgrading the Oracle Enterprise Session Border Controller software:

1. Obtain the name and location of the target software image file from either Oracle Software Delivery Cloud (*https://edelivery.oracle.com/*) or My Oracle Support (*https://support.oracle.com*) as applicable.
2. The Acme Packet 4500, Acme Packet 6100, and Acme Packet 6300 should be provisioned with the 64-bit Oracle Enterprise Session Border Controller image file in the boot parameters. 64-bit image files are recognized by the "64" between the image revision and file extension. e.g., nnSCZ720.64.bz . The Acme Packet 3820 should be provisioned with a 32-bit Oracle Enterprise Session Border Controller image file in the boot parameters. 32-bit image files are recognized by the "32" between the image revision and file extension. e.g., nnSCZ720.32.bz .
3. Verify the integrity of your configuration using the ACLI verify-config command.

4. Back up a well-working configuration. Name the file descriptively so you can fall back to this configuration easily.
5. If your hardware is the Acme Packet 3820 or Acme Packet 4500 verify that the stage 1 and stage 2 bootloaders are dated July 3, 2013 or later. Use the show version boot ACLI command for this query. Stage 1 and stage 2 bootloaders are available from My Oracle Support (*https://support.oracle.com*) under their respective hardware listing.
6. Refer to the Oracle Enterprise Session Border Controller Release Notes for any caveats involving software upgrades.
7. Note that all Oracle Enterprise Session Border Controllers have password secure mode enabled-meaning that you must accurately track your password information. To learn more about password secure mode, refer to the Maintenance and Troubleshooting Guide's Data Storage Security section

AP3820 and AP4500 Upgrade Requirement

To run release Version E-CZ7.1.0 on an AP3820 or AP4500, the Stage1 and Stage2 bootloaders MUST be dated July 3, 2013 or later. See the Oracle Enterprise Session Border Controller Release Notes guide for more information. Use the **show version boot** command to confirm this.

```
ACMEPACKET# show version boot
Bootloader Info
---------------
Stage 1: Jul  3 2013 13:16:30
Stage 2: Jul  3 2013 13:16:30
```

Stand-alone Upgrade

This process incurs system downtime; your Oracle Enterprise Session Border Controller stops passing traffic for a period of time. Please plan for your standalone upgrade accordingly.

Boot Loader Requirements

AP3820 and AP4500 Boot Loaders

The AP 3820 and AP 4500 require Stage 1, Stage 2, and Stage 3 boot loaders.

Stage 1 and Stage 2 boot loaders should be dated no earlier than July 3, 2013. (MOS patch # 18185632) Use the **show version boot** command to view current boot loader version on your system.

Stage 1 and Stage 2 boot loader updates are available on My Oracle Support, listed under the respective hardware.

The Stage 3 boot loader accompanies the OCSBC image file, as distributed. Install the image file according to the instructions in the Maintenance and Troubleshooting Guide.

Check /boot for free space

On the Oracle Enterprise Session Border Controller, check for adequate space in the /boot volume to upload the new boot image and bootloader. Use the check-space-remaining boot command.

```
ACMEPACKET# check-space-remaining boot
code: 24759488/25760512 bytes (99%) remaining
ACMEPACKET#
```

You may delete files from an SFTP client if you need to free space.

Image and Boot Loader File Conventions

The AP4500 is provisioned with the 64-bit Oracle Enterprise Session Border Controller image file in the boot parameters. 64-bit image files are recognized by the "64" between the image revision and file extension. e.g., nnECZ710.64.bz .

The AP3820 is provisioned with a 32-bit Oracle Enterprise Session Border Controller image file in the boot parameters. 32-bit image files are recognized by the "32" between the image revision and file extension. e.g., nnECZ710.32.bz .

All platforms require that you install a stage 3 boot loader. The stage 3 boot loader is identified by the .boot file extension. The stage 3 boot loader and system image file have identical name portions of the filename, and are distributed together. For this software, the GA system image and stage 3 boot loader are nnecz710.64.bz and nnECZ710.boot,respectively.

Update the Stage 3 Boot Loader and System Image

Whenever you upgrade the software image, upload the stage 3 boot loader image file and reboot the system with the new system software image. The stage 3 boot loader is backward compatible as well, so you can still boot into older software images with a newer stage 3 boot loader.

To ensure compatibility, copy the stage 3 boot loader to /boot/bootloader before you update the boot parameters to use the new software image file. The boot loader file must be renamed to /boot/bootloader on the target system with no file extension. When upgrading an HA pair, you must perform the upgrade procedure on each HA node.

Procedure

1. Obtain the stage 3 boot loader image file (*.boot).
2. Upload the stage 3 boot loader image file (*.boot) as /boot/bootloader to your system using an SSH File Transfer Protocol (SFTP) client.
3. Upload the new system software image (*.bz) to /boot/.

```
[Downloads]$ ls -la
total 148820
drwxr-xr-x  2 bob src     4096 Jun 17 15:16 .
drwxr-xr-x 28 bob src     4096 May 21 14:17 ..
-rw-r--r--  1 bob src 10164527 Jun 17 15:15 nnSCZ720.64.boot
-rw-r--r--  1 bob src 73849839 Jun 17 15:15 nnSCZ720.64.bz
[Downloads]$ sftp user@172.30.46.20
user@172.30.46.20's password:
Connected to 172.30.46.20.
sftp> cd /boot
sftp> put nnSCZ720.64.boot
Uploading nnSCZ720.64.boot to /boot/nnSCZ720.64.boot
nnSCZ720.64.boot                              100% 9926KB   9.7MB/
s   00:01
sftp> rename nnSCZ720.64.boot bootloader
sftp> put nnSCZ720.64.bz
Uploading nnSCZ720.64.bz to /boot/nnSCZ720.64.bz
nnSCZ720.64.bz                                100%   70MB  14.1MB/
s   00:05
sftp> bye
Received disconnect from 172.30.46.20: 11: Logged out.
[Downloads]$
```

Next Steps

Set the boot parameters to boot the new system software image, and reboot the system.

Software Upgrade Procedure

The following procedure describes how to upgrade a Oracle Enterprise Session Border Controller with a new software image. In this procedure, the image file is located on the Oracle Enterprise Session Border Controller's local file system in /boot.

To upgrade a software image on a stand-alone system:

1. Change the boot configuration parameters to use the new image.

Scroll through the boot parameters by pressing Enter. Stop when you reach the file name boot parameter and type the appropriate file name next to the previous file name. Press <Enter> to continue scrolling through the boot parameters.

The following example uses the filenames /code/images/nnSCX640.gz and /boot/nnSCZ712.64.bz.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device          : eth0
processor number     : 0
host name            : boothost
file name            : /code/images/nnSCX640.gz /code/images/nnSCZ712.64.bz
```

2. Reboot the Oracle Enterprise Session Border Controller using the reboot command.

   The Oracle Enterprise Session Border Controller should now be successfully running the new release.

HA Upgrade

In the descriptions and processes outlined below, E-SBC1 is initially the standby system and E-SBC2 is initially the active system. Please read the following procedures carefully before beginning the upgrade. If necessary, you can back out of the upgrade once during the upgrade procedure and once after you have completed the upgrade procedure.

Upgrade Process

To upgrade a software image for an HA node:

1. Confirm that Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 start up and are synchronized.

   You must also make sure that all of the running and current configurations on Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2 have the same number. In the following examples, all of the configuration versions are 5.

   On Oracle Enterprise Session Border Controller1 and Oracle Enterprise Session Border Controller2, use the ACLI show health command to make sure that all processes are synchronized.

   On Oracle Enterprise Session Border Controller1, show the current configuration version by using the ACLI **display-current-cfg-version** command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its current configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

   ```
   NETNETSBC1# display-current-cfg-version
   Current configuration version is 5
   NETNETSBC1#
   NETNETSBC2# display-current-cfg-version
   Current configuration version is 5
   NETNETSBC2#
   ```

   On Oracle Enterprise Session Border Controller1, show the running configuration version by using the ACLI **display-running-cfg-version** command. Then use the same command on Oracle Enterprise Session Border Controller2 and be sure that its running configuration version is the same as the one on Oracle Enterprise Session Border Controller1.

   ```
   NETNETSBC1# display-running-cfg-version
   Running configuration version is 5
   NETNETSBC1#
   NETNETSBC2# display-running-cfg-version
   Running configuration version is 5
   NETNETSBC2#
   ```

2. On Oracle Enterprise Session Border Controller1, before loading the software image to the flash, check the remaining space in the /boot directory using the ACLI **check-space-remaining code** command.

```
NETNETSBC1# check-space-remaining boot
boot: 24759488/25760512 bytes (99%) remaining
NETNETSBC1#
```

If you see less than 50% of the memory remaining, delete older stored firmware images to make space.

At a minimum, we recommend that you leave the diags.gz file and the currently running release on the flash memory (in the event that a rollback is required).

3. Upload the Oracle Enterprise Session Border Controller software image file and stage three bootloader to the /boot directory using an SFTP client. See: *Upgrade the Stage 3 Bootloader*.
4. Change the boot configuration parameters on Oracle Enterprise Session Border Controller1 to use the appropriate new release software image.

   Note: From the point that you upgrade the image file, do not make any configuration changes. Likewise, do not use the save-config or activate-config commands. Once you execute the save-config command, the configuration can not be guaranteed to be backward compatible should you have to back out of the upgrade.

   Access the boot parameters on Oracle Enterprise Session Border Controller1:

   - In the ACLI configure terminal menu, type **bootparam** and press <Enter> to being displaying the list of boot parameters.

     Scroll through the boot parameters by pressing <Enter>. Stop when you reach the file name boot parameter.

     The following example uses the filenames /code/images/nnSCX640.gz and /boot/nnSCZ712.64.bz.

     ```
     NETNETSBC1# configure terminal
     NETNETSBC1(configure)# bootparam
     '.' = clear field;  '-' = go to previous field;  ^D = quit
     boot device          : eth0
     processor number     : 0
     host name            : boothost
     file name            : /code/images/nnSCX640.gz /boot/nnSCZ712.64.bz
     ```

     As shown above, type the new Release file name next to the previous one, including the path. Press <Enter> to continue scrolling through the boot parameters.

     Reboot Oracle Enterprise Session Border Controller1.
5. After Oracle Enterprise Session Border Controller1 has completed the boot process, use the verify-config command to confirm that the configuration has been upgraded properly.

   ```
   NETNETSBC1# verify-config
   ```
6. Confirm the Oracle Enterprise Session Border Controller1 is running the new boot image using the show version command.

   ```
   NETNETSBC1# show version
   Acme Packet Net-Net 4500 SCZ7.1.2 GA (Build 165)
   Build Date=07/17/13
   NETNETSBC1#
   ```
7. Use the ACLI show health command to confirm that Oracle Enterprise Session Border Controller1 is the standby system.
8. As you did for Oracle Enterprise Session Border Controller1, configure the boot parameters on Oracle Enterprise Session Border Controller2 so to use the new Net-Net Release S-CX6.4.0 software image. Then reboot Oracle Enterprise Session Border Controller2.

   ```
   NETNETSBC2# reboot
   ------------------------------------------------------
   WARNING: you are about to reboot this SD!
   ------------------------------------------------------
   Reboot this SD [y/n]?: y
   ```

   Rebooting Oracle Enterprise Session Border Controller2 causes Oracle Enterprise Session Border Controller1 to become the active system in the HA node.

9. When Oracle Enterprise Session Border Controller2 is finished rebooting, use the ACLI show health command to confirm that it is in the standby state.

Note: If you need to revert to older image, use the HA Backout Procedure.

HA Backout Procedure

If you reach the point in your upgrade procedure where you have upgraded both Oracle Enterprise Session Border Controllers in the HA node to a later release that you decide you no longer want to use, you can fall back to a previous release. This section shows you how to fall back to an older image with both systems in your HA node upgraded.

In the descriptions and processes outlined below, Oracle Enterprise Session Border Controller1 is the active system and Oracle Enterprise Session Border Controller2 is the standby system. The procedure uses these designations because when you have completed upgrade process specific to these releases, Oracle Enterprise Session Border Controller1 is the active system.

To backout to a previous (older) release with the both Oracle Enterprise Session Border Controllers in the HA node upgraded:

1. Change the boot parameters on Oracle Enterprise Session Border Controller2 to use the appropriate Release SCX6.4.0 software image.

   Using one of these methods, access the boot parameters on Oracle Enterprise Session Border Controller2:

   - Reboot the Oracle Enterprise Session Border Controller using any of the ACLI reboot commands. Stop the booting process by hitting the Space bar on your keyboard to halt boot-up when you see this message: Press any key to stop auto-boot.... Type a c and press Enter to begin displaying the boot parameters.
   - In the ACLI configure terminal menu, type bootparam and press Enter to being displaying the list of boot parameters.

     Scroll through the boot parameters by pressing Enter. Stop when you reach the file name boot parameter.

     The following example uses the filenames /code/images/nnSCX640.xz and /boot/nnSCZ712.64.bz.

```
ACMEPACKET2# configure terminal
ACMEPACKET1(configure)# bootparam
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device          : eth0
processor number     : 0
host name            : boothost
file name            : /boot/nnSCZ712.64.bz /code/images/nnSCX640.xz
```

     In the example above, type the appropriate Release S-CX6.4.0 file name next to the Release S-CZ7.1.2 file name. Press <Enter> to continue scrolling through the boot parameters.

     Exit to the main Superuser prompt.

```
ACMEPACKET2(configure)# exit
ACMEPACKET2#
```

2. Reboot Oracle Enterprise Session Border Controller2.
3. Using the ACLI show version command to confirm that you are using the appropriate release.

```
ACMEPACKET2# show version
ACME PACKET 4500 Firmware S-CX6.4.0 GA
07/15/10
ACMEPACKET2#
```

4. Initiate a switchover on Oracle Enterprise Session Border Controller2.

```
ACMEPACKET2# notify berpd force
```

   At this point, Oracle Enterprise Session Border Controller2 becomes the active system running Release S-CX6.4.0. Oracle Enterprise Session Border Controller1 is now the standby system running Release S-CZ7.1.2

5. On Oracle Enterprise Session Border Controller1, change the boot parameters as you did in Step 1 of this procedure.
6. Reboot Oracle Enterprise Session Border Controller1.

Moving a Configuration

This section outlines a process for moving an existing Oracle Enterprise Session Border Controller configuration to a new system. Process summary:

1. Create a backup configuration file on the source Oracle Enterprise Session Border Controller.
2. Using SFTP, copy the source backup from the source to the destination Oracle Enterprise Session Border Controller .
3. Restore the newly-transferred backup on the target Oracle Enterprise Session Border Controller.

Backup Commands

The Oracle Enterprise Session Border Controller software includes a set of commands for easily working with backup configurations. These commands are backup-config, display-backups, delete-backup-config, restore-backup-config.

To back up a configuration, use the backup-config command. You can confirm that your backup has been created with the display-backups command. When the backup-config command is executed, the system checks for sufficient resources to complete the operation. If resources are sufficient, the system creates the backup. If resources are insufficient, the task is not completed and the system displays the limiting resources and recommends completing the task at another time.

Backups are created as gzipped tar files in a .tar.gz format. They are stored in the /code/bkups directory on the Acme Packet 4000.

Backing up the current configuration

To create a backup:

In superuser mode, use the **backup-config** command followed by a descriptive filename for the backup you are creating.

```
ACMEPACKET#backup-config 02_Feb_2008
task done
ACMEPACKET#
```

Listing Backups

You can view the backups available on your system using the **display-backups** command.

To list available backup configurations:

In Superuser mode, enter the **display-backups** command. A list of available backup files from the /code/bkups directory is displayed on the screen.

```
ACMEPACKET# display-backups
test_config.tar.gz
test-config.tar.gz
runningcfgtest.tar.gz
runningtest_one.tar.gz
BACK_UP_CONFIG.tar.gz
02_Feb_2008.tar.gz
01_Feb_2008.tar.gz
ACMEPACKET#
```

Copy the Backup to the destination Oracle Enterprise Session Border Controller

Send the backup configuration file by way of SFTP from the source to destination Oracle Enterprise Session Border Controller.

To copy a backup configuration from the source to destination Oracle Enterprise Session Border Controller:

1. Use an SFTP client to connect to the Acme Packet 4250 using the default username: user and password: acme. The management IP address is configured in the bootparams.
2. Change directory to where you want to upload a file.
   - cd /code/bkups for backup configurations
3. Type bin and press Enter to force the SFTP program into binary mode.
4. Upload the file you want to transfer by typing the filename and pressing Enter.

```
C:\Documents and Settings>ftp 172.30.55.127
Connected to 172.30.55.127.
220 VxWorks (1.0) FTP server ready
User (172.30.55.127:(none)): user
331 Password required
Password:
230 User logged in
ftp> cd /code/bkups
250 Changed directory to "/code/bkups"
ftp> bin
200 Type set to I, binary mode
ftp> put 02_Feb_2008.tar.gz
200 Port set okay
150 Opening BINARY mode data connection
226 Transfer complete
ftp: 9587350 bytes sent in 51.64Seconds 185.65Kbytes/sec.
ftp>
```

Restoring Backups

To restore a backup configuration on the Oracle Enterprise Session Border Controller:

1. In Superuser mode, enter the restore-backup-config command followed by the backup filename you want to restore to the current configuration. You must explicitly name the backup file you want to restore, including the file extension

```
ACMEPACKET4500# restore-backup-config 02_Feb_2008.tar.gz
Need to perform save-config and activate/reboot activate for changes to
take effect...
task done
ACMEPACKET4500#
```

2. Correct the Virtual MAC address configuration established on the former device to be suitable for the new device.

   Establish the base MAC needed for HA operation by, first, determining the base MAC by way of the ethernet address value of the show media physical command.

```
ACMEPACKET4500#show media physical
s0p0 (media slot 0, port 0)
     Flags: UP BROADCAST MULTICAST ARP RUNNING
     Type: ETHERNET_CSMACD
     Admin State: enabled
     Auto Negotiation: enabled
…
     Ethernet address is 00:08:25:01:08:44
```

   Apply the formula for calculating virtual MAC addressing to the MAC addressing used for the Acme Packet 4500 system. This formula is described in the Oracle Enterprise Session Border Controller ACLI Configuration Guide.

Configure the physical interfaces with the computed virtual MAC addressing. Refer to the following command line sequence as an example of this procedure.

```
ACMEPACKET4500# configure terminal
ACMEPACKET4500(configure)# system
ACMEPACKET4500(system)# phy-interface
ACMEPACKET4500(phy-interface)# select
<name>:
1: s0p0
2: s1p0
selection: 1
ACMEPACKET4500(phy-interface)# virtual-mac 00:08:25:01:08:48
ACMEPACKET4500(phy-interface)# done
phy-interface
        name                            s0p0
        operation-type                  Media
        port                            0
        slot                            0
        virtual-mac                     00:08:25:01:08:48
```

3. Save the configuration.

```
ACMEPACKET4500# save-config
```

4. Activate the configuration.

```
ACMEPACKET4500# activate-config
```

6

System Configuration

This chapter explains how to configure system-level functionality for the Oracle Enterprise Session Border Controller. Both physical and network interfaces as well as general system parameters are required to configure your Oracle Enterprise Session Border Controller for service. Accounting functionality, SNMP configurations, trap configurations, and host routes are optional.

The following configurations are explained in this chapter:

- General system parameters—used for operating and identification purposes. In general, the informational fields have no specific effect on services, but are important to keep populated. The default gateway parameter is included here. It requires special attention since its configuration is dependent on the type of traffic the Oracle Enterprise Session Border Controller is servicing.
- Physical and network interfaces—enables the Oracle Enterprise Session Border Controller to communicate with any network element. Interfaces are one of the most basic configurations you need to create.
- SNMP—used for monitoring system health throughout a network.
- Syslogs and Process logs—used to save a list of system events to a remote server for analysis and auditing purposes.
- Host routes—used to instruct the Oracle Enterprise Session Border Controller host how to reach a given network that is not directly connected to a local network interface.

General System Information

This section explains the parameters that encompass the general system information on a Oracle Enterprise Session Border Controller.

System Identification

Global system identification is used primarily by the Oracle Enterprise Session Border Controller to identify itself to other systems and for general identification purposes.

Connection Timeouts

It is important to set administrative session timeouts on the Oracle Enterprise Session Border Controller for security purposes. If you leave an active configuration session unattended, reconfiguration access is left open to anyone. By setting a connection timeout, only a short amount of time needs to elapse before the password is required for Oracle Enterprise Session Border Controller access.

Timeouts determine the specified time period that must pass before an administrative connection is terminated. Any subsequent configuration activity can only be performed after logging in again to the Oracle Enterprise Session Border Controller. The timeout parameter can be individually specified for Telnet sessions and for console port sessions.

After the Telnet timeout passes, the Telnet session is disconnected. You must use your Telnet program to log in to the Oracle Enterprise Session Border Controller once again to perform any further configuration activity.

After the console timeout passes, the console session is disconnected. The current session ends and you are returned to the login prompt on the console connection into the Oracle Enterprise Session Border Controller.

Configuring General System Information

This section explains how to configure the general system parameters, timeouts, and the default gateway necessary to configure your Oracle Enterprise Session Border Controller.

ACLI Instructions and Examples

To configure general system information:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type system-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

The following is an example what a general system information configuration might look like. Parameters not described in this section are omitted below.

```
ACMEPACKET(system-config)# show
system-config
        hostname                        test1
        description                     Example SD
        location                        Row 3, Rack 4, Slot 451
        default-gateway                 10.0.2.1
        telnet-timeout                  1000
        console-timeout                 1000
        last-modified-date              2004-12-08 20:15:43
```

When showing a single-instance configuration element such as system-config, you must first use the select command to select the configuration element prior to viewing.

System Identification

You must specify identification parameters for this Oracle Enterprise Session Border Controller.

Set the following parameters to configure the system identification:

1. hostname—Set the primary hostname used to identify the system. This parameter is used by the software for informational purposes.
2. description—Enter a textual description of the system. This parameter is used for informational purposes.
3. location—Set a location description field for the system. This parameter is used for informational purposes. For example, you might include the site name and address of the location where the system chassis is located.
4. default-gateway—Set the default gateway for this Oracle Enterprise Session Border Controller. This is the egress gateway for traffic without an explicit destination. The application of the Oracle Enterprise Session Border Controller determines the configuration of this parameter.

Hostname Field Lengths

This feature applies to the Linux OS for the Net-Net Enterprise Session Director Server Edition (NN-ESD SE) and the Net-Net Enterprise Session Director Virtual Machine Edition (NN-ESD VME) platforms only.

The NN-ESD boot loader and application target/host name fields accept up to 63 ASCII characters (NULL not included). It modifies the ACLI command, configure terminal bootparam and allows you to enter a hostname of up to 63 characters. Previously, these fields on the Net-Net 3280 and 4500 accepted only 24 ASCII characters and the name was truncated if you specified more than 24 characters for the hostname. For example, if the target name was:

alpha-bravo-charlie-1234567890 (30 characters)

the ACLI on previous systems displayed:

```
alpha-bravo-charlie-1234(configure)#
```

It now displays the first 12 characters + 3 periods + last 12 characters (NULL not included) as follows:

```
alpha-bravo-...e-1234567890(configure)#
```

You must upgrade the NN-ESD boot loader for this feature to take affect. For more information about upgrading the boo tloader, see Chapter 2, the section, *Bootloader Upgrade Procedure (5)*.

Configuring Connection and Debug Logging Timeouts

Configure the timeouts for terminal sessions on this Oracle Enterprise Session Border Controller. These parameters are optional.

Set the following parameters to configure the connection timeouts:

1. telnet-timeout—Set the Telnet timeout to the number of seconds you want the Oracle Enterprise Session Border Controller to wait before it disconnects a Telnet session or an SSH session. The default value is 0. The valid range is:
   - Minimum—0
   - Maximum—65535
2. console-timeout—Set the console timeout to the number of seconds you want the Oracle Enterprise Session Border Controller to wait before it ends the console session. The default value is 0. The valid range is:
   - Minimum—0
   - Maximum—65535
3. debug-timeout—Set the time in seconds you want to use for the debug timeout. This is the time allowed before the Oracle Enterprise Session Border Controller times out log levels for system processes set to debug using the ACLI notify and debug commands.

   This command does not affect log levels set in your configuration (using parameters such as system-config>process-log-level) or those set using the ACLI log-level command.

   The valid range is:
   - Minimum—0
   - Maximum—65535

ACLI Command-Line Tools

This section describes some tools you can use to enhance your experience when using the ACLI. For ACLI behaviors when using specific commands, see Chapter 4, the section, *ACLI Expected Behavior (118)*.

Tab Completion

Tab completion is available for every boolean, enumeration, and reference parameter. A reference parameter is one whose value references a value in another configuration. For example, a SIP interface configuration might reference the name parameter of a realm configuration; tab completion show you the names of the realms from which you can choose.

Consistent and Detailed Error Messages

Error message the system produces and displays in the ACLI are consistent and detailed. If your entry is invalid, the system tells you the range for a valid value.

```
ACMEPACKET(system-config)# debug-timeout 99999
% Invalid Input
    Value is not in the range 0..65535
```

Question Mark Help

When you enter a parameter name and then a question mark, the ACLI shows valid ranges and options for values, and default values when they apply.

```
ACMEPACKET(system-config)# debug-timeout ?
<0..65535>  Time before debugging is disabled (in seconds)
            Zero (0) disables timeouts
            Default: 0
```

Password Entry

Password entry is the same for all types of passwords you can enter on the Oracle Enterprise Session Border Controller. There are two methods of password entry:

- You can use the parameter name with the mypass argument to enter your password directly into the system.

  ```
  ACMEPACKET(snmp-user-entry)# auth-password mypass
  ```

- You can use the parameter name and press Enter to access the Enter Password: prompt; from there, you can type your password, which does not echo on the screen. After you enter your password at this prompt, the Retype Password:appears and you can retype your password, which does not echo on the screen.

  ```
  ACMEPACKET(snmp-user-entry)# auth-password
  Enter password: [Enter your password here and press Enter.]
  Retype password: [Re-enter your password here and press Enter.]
  Password updated
  ```

When you display configuration information containing passwords, only a series of asterisks (*) appears.

```
ACMEPACKET(snmp-user-entry)# show
snmp-user-entry
        user-name
        auth-password                           ********
        priv-password
```

Find Commands

Using the set of ACLI find commands, you can search the Oracle Enterprise Session Border Controller's configuration or the ACLI menu for configuration information. You enter search for this information using a specific string value for the configuration you want to find. The command searches the editing configuration, running configuration, and ACLI command menu for the string value.

To prevent unwieldy result displays, your searches are limited to 100 results. If your search extends beyond 100, the system shows the first 100 and then displays this message: Number of results exceeds limit.

Unlicensed configuration is not displayed.

```
ACMEPACKET# find [configuration|running-config|command] [attribute] <string>
```

- Configuration, running config, command—You can specify the area you want to search using these arguments. This argument is optional. Without it, the find command searches all three areas for the specified criteria; see *Searching Globally*.
  - configuration—Searches the editing configuration.
  - running-config—Seaches the running configuration.
  - command—Searches the ACLI command menu. See *Searching the ACLI Menu*.

- Attribute—When searching editing and running configurations, you can target your search to specific configuration attributed by entering an ACLI attribute name. The attribute can provide additional context for searching for the string you enter with this command. This argument is optional. Note that if you carry out the command without an attribute being specified, the output provides contextual information in the form of the configuration object and attribute where the string value appears.

```
ACMEPACKET# find configuration registration-caching enabled
session-router -> sip-interface [net172]
    registration-caching enabled
session-router -> sip-interface [net192]
    registration-caching enabled
Found 2 instances
ACMEPACKET#
```

- String—The search criteria are specified by the string value you tell the Oracle Enterprise Session Border Controller to find, and this value is required. Searches are case-sensitive, and support sub-string matching. Note that the displayed information includes the string itself and the configuration object's location.

```
ACMEPACKET# find running-config 172.16
session-router -> local-policy [*; *; 172.16.11.2]
    to-address 172.16.11.2
session-router -> local-policy [*; *; 172.16.11.2] -> local-policy-
attribute [net172; 172.16.11.2]
    next-hop 172.16.11.2
session-router -> local-policy [*; *; 123] -> local-policy-attribute
[net172; 172.16.7.201]
    next-hop 172.16.7.201
session-router -> local-policy [net192; *; *] -> local-policy-attribute
[net172; 172.16.30.100]
    next-hop 172.16.30.100
system -> network-interface [M10:0]
    ip-address 172.16.101.11
    pri-utility-addr 172.16.11.6
    sec-utility-addr 172.16.11.7
    gateway 172.16.11.1
    hip-ip-list 172.16.101.11
    icmp-address 172.16.101.11
session-router -> sip-interface [net172] -> sip-port [172.16.101.11:5060/
UDP]
    address 172.16.101.11
media-manager -> steering-pool [172.16.101.11=15000+net172]
    ip-address 172.16.101.11
Found 12 instances
ACMEPACKET#
```

Quit Command in Configuration Mode

When using the "quit" command after saving a configuration, the system exits the configuration mode and displays the root prompt for the Oracle Enterprise Session Border Controller.

In the following example, the "quit" command is used and the session-recording-server configuration is saved. The root prompt of ACMEPACKET displays.

```
ACMEPACKET(session-recording-server)# quit
Save Changes [y/n]?: y
session-recording-server
        name                                    crs2
        description
        realm                                   2.2.2.2
        mode                                    selective
        destination                             3.3.3.3
        port                                    5060
        transport-method                        DynamicTCP
        ping-method
        ping-interval                           0
```

```
        last-modified-by                        admin@10.1.25.17
        last-modified-date                      2013-06-06 11:49:24
ACMEPACKET#
```

Searching the ACLI Menu

When you use the find command option, the systems displays all menu items matching the string you specify (case-sensitive, sub-string match). This search includes all root-level and configuration-level parameters, and displays the full path to each value it displays.

```
ACMEPACKET# find command srtp
(root) show sa stats <srtp>
(configure) media-manager realm-config srtp-msm-passthrough
(configure) media-manager vbg-config srtp
(configure) security media-security sdes-profile srtp-auth
(configure) security media-security sdes-profile srtp-encrypt
(configure) security security-config srtp-msm-password
(configure) security security-config srtp-msm-attr-name
Found 7 instances
ACMEPACKET#
```

Searching Globally

When you use the find command without directing your search to the editing configuration, running configuration, or ACLI command (parameter) menu, the system searches and returns data for all three. The display identifies in which of the three areas results are found.

Note that:

- The system does not display editing configuration if it is the same as running configuration.
- Command results are not displayed if the Attribute context is present.

```
ACMEPACKET# find *
Command menu --------------------------
(root) capture start realm <main-filter> *
(root) capture start session-agent <main-filter> *
(root) capture start global *
(root) show directory *
Editing configuration --------------------
session-router -> local-policy [*; *; 172.16.11.2]
    from-address *
    source-realm *
session-router -> local-policy [*; *; 192.168.11.2]
    from-address *
    source-realm *
session-router -> local-policy [*; *; 123]
    from-address *
    source-realm *
session-router -> local-policy [net172; *; *]
    from-address *
    to-address *
session-router -> local-policy [net192; *; *]
    from-address *
    to-address *
session-router -> sip-config
    registrar-domain *
    registrar-host *
session-router -> sip-monitoring
    monitoring-filters *,
Running configuration ------------------
session-router -> local-policy [*; *; 172.16.11.2]
    from-address *
    source-realm *
session-router -> local-policy [*; *; 192.168.11.2]
    from-address *
```

```
    source-realm *
session-router -> local-policy [*; *; 123]
    from-address *
    source-realm *
session-router -> local-policy [net172; *; *]
    from-address *
    to-address *
session-router -> local-policy [net192; *; *]
    from-address *
    to-address *
session-router -> sip-config
    registrar-domain *
    registrar-host *
session-router -> sip-monitoring
    monitoring-filters *,
Found 30 instances
ACMEPACKET#
```

Displaying Copying and Pasting Configurations

These commands allow you to handle your Oracle Enterprise Session Border Controller configuration in the following ways:

- Displaying a summary, abbreviated version of a configuration
- Copying and pasting pre-existing values into your configuration

Abbreviated Configuration Summary

Adding the short argument to the end of any show config or show running config command instructs the system to display an abbreviated output. The short argument tells the system to display only those parameters whose values have changed from the defaults.

```
ACMEPACKET# show run sip-interface short
sip-interface
    realm-id                                net172
    sip-port
        address                                 172.16.101.11
    registration-caching                    enabled
sip-interface
    realm-id                                net192
    sip-port
        address                                 192.168.101.11
    registration-caching                    enabled
```

Configuration Pasting

Using paste-config, you can paste the text output of a show run show config or show running-config (i.e., show run) command into the ACLI. The paste-config command is part of the configure terminal menu. The text you enter must match exactly the output of the show config/show run command, including any indentation.

After the pasting process, the system checks for configuration errors and asks you for acceptance for the pasted configuration. Accepting the configuration is the same as using the ACLI done command or saving the pasted information to the editing cache for each configuration object. You can then edit these objects using the configuration menus, or you can save and activate the configuration.

You cannot paste:

- Password fields
- Privilege
- Unlicensed features

To paste copied configuration into the ACLI:

1. Access the ACLI's configure terminal branch and type paste-config.

```
ACMEPACKET# conf t
ACMEPACKET(configure)# paste-config
```

2. Paste in the configuration data you want the system to accept, including any indentations. To stop data entry, enter <CTRL-D>.

```
Paste configuration onto console.  Enter <CTRL-D> to stop.
sip-interface
    realm-id                                net172
    sip-port
        address                                 172.16.101.11
    registration-caching                    enabled
sip-interface
    realm-id                                net192
    sip-port
        address                                 192.168.101.11
    registration-caching                    enabled
-----------------
(0 errors)
```

3. At the system prompt, press y to continue with accepting the configuration or press n to terminate the process. The system confirms your action.

```
Do you want to accept this configuration [y/n]?: y
(2 top-level objects written)
ACMEPACKET(configure)#
```

During the pasting process, the system performs error checking. Error checking includes identification of:

- Unrecognized tokens
- Invalid values for an a parameter based on the parameter's constraints
- Proper value checking
- Proper licensing

Pasted configuration that returns errors looks like this example process, in which the user has terminated the pasting process.

```
ACMEPACKET# conf t
ACMEPACKET(configure)# paste-config
Paste configuration onto console.  Enter <CTRL-D> to stop.
sip-interface
    foobar                                  net172
    sip-port
        address                                 172.16.101.11
    registration-caching                    enabled
sip-interface
    realm-id                                net192
    sip-port
        address                                 dot.com
    registration-caching                    dot.com
dot.com
-----------------
Failed to get dot.com
% sip-interface []
  Unknown attribute: dot.com
% sip-interface []
  Save error:
Realm-id must be set
% sip-interface [net192] > sip-port [:5060/UDP]
  Could not set attribute address to "dot.com" : Invalid Input
    Value not a valid IP address
% sip-interface [net192]
  Could not set attribute registration-caching to "dot.com" : Invalid Input
    Value must be either enabled or disabled
% Unknown object: dot.com
% (root level)
```

```
  Unknown object: dot.com
(6 errors)
Do you want to accept this configuration [y/n]?: n
Operation cancelled
ACMEPACKET(configure)#
```

Timezones

The timezone on the Oracle Enterprise Session Border Controller must be set manually via the ACLI using one of two methods:

- using the timezone-set command at the root prompt. This commands starts a timezone wizard that allows you to answer prompts specifically related to timezone settings. You can set your timezone location and the wizard automatically sets the daylight savings time for the location you select.
- at the path system->timezone. This parameter allows you to create a timezone name and apply specific instructions for daylight savings time (DST) and specify the number of minutes from Coordinated Universal Time (UTC). If you initiated the timezone-set wizard previous to accessing this parameter, the settings for system->timezone are already populated. You can change them if required.

It is recommended you set the timezone after first boot of the system.

About UTC Timezones

Coordinated Universal Time (UTC) is used as the official world reference for time. Coordinated Universal Time replaced the use of Greenwich Mean Time (GMT) in 1972. Sometimes time zones are represented similar to UTC - 5h or GMT - 5h. In this example, the (-5h) refers to that time zone being five hours behind UTC or GMT and so forth for the other time zones. UTC +5h or GMT +5h would refer to that time zone being five hours ahead of UTC of GMT and so forth for the other time zones.

The usage of UTC and GMT is based upon a twenty four hour clock, similar to military time, and is based upon the 0° longitude meridian, referred to as the Greenwich meridian in Greenwich, England.

UTC is based on cesium-beam atomic clocks, with leap seconds added to match earth-motion time, where as Greenwich Mean Time is based upon the Earth's rotation and celestial measurements. UTC is also known as Zulu Time or Z time.

In areas of the United States that observe Daylight Saving Time, local residents move their clocks ahead one hour when Daylight Saving Time begins. As a result, their UTC or GMT offset would change from UTC -5h or GMT - 5h to UTC -4h or GMT - 4h. In places not observing Daylight Saving Time the local UTC or GMT offset will remain the same year round. Arizona, Puerto Rico, Hawaii, U.S. Virgin Islands and American Samoa do not observe Daylight Saving Time.

In the United States Daylight Saving Time begins at 2:00 a.m. local time on the second Sunday in March. On the first Sunday in November areas on Daylight Saving Time return to Standard Time at 2:00 a.m. The names in each time zone change along with Daylight Saving Time. Eastern Standard Time (EST) becomes Eastern Daylight Time (EDT), and so forth. A new federal law took effect in March 2007 which extends Daylight Saving Time by four weeks.

The United States uses nine standard time zones. From east to west they are Atlantic Standard Time (AST), Eastern Standard Time (EST), Central Standard Time (CST), Mountain Standard Time (MST), Pacific Standard Time (PST), Alaskan Standard Time (AKST), Hawaii-Aleutian Standard Time (HST), Samoa standard time (UTC-11) and Chamorro Standard Time (UTC+10). The following tables identify the standard time zone boundaries and the offsets.

Standard Timezone Boundaries Table

Coordinated Universal Time (UTC)	Greenwich Mean Time (GMT)
UTC/GMT +0	UTC/GMT +0

Timezone Offsets Table

United States GMT/UTC Offsets			
Time Zone in United States	Examples of places in the United States using these Time Zones	UTC Offset Standard Time	UTC Offset Daylight Saving Time
Atlantic	Puerto Rico, US Virgin Islands	UTC - 4h	N/A
Eastern	Connecticut, Delaware, Florida, Georgia, part of Indiana, part of Kentucky, Maine, Maryland, Massachusetts, Michigan, New Hampshire, New Jersey, New York, North Carolina, Ohio, Pennsylvania, Rhode Island, South Carolina, part of Tennessee, Vermont, Virginia and West Virginia	UTC - 5h	UTC - 4h
Central	Alabama, Arkansas, Florida, Illinois, part of Indiana, Iowa, part of Kansas, part of Kentucky, Louisiana, part of Michigan, Minnesota, Mississippi, Missouri, Nebraska, North Dakota, Oklahoma, part of South Dakota, part of Tennessee, most of Texas, and Wisconsin	UTC - 6h	UTC - 5h
Mountain	Arizona*, Colorado, part of Idaho, part of Kansas, Montana, part of Nebraska, New Mexico, part of North Dakota, part of Oregon, part of South Dakota, part of Texas, Utah, and Wyoming	UTC - 7h	UTC - 6h * n/a for Arizona
Pacific	California, part of Idaho, Nevada, most of Oregon, Washington	UTC - 8h	UTC - 7h
Alaska	Alaska and a portion of the Aleutian Islands that is east of 169 degrees 30 minutes west longitude observes the Alaska Time Zone.	UTC - 9h	UTC - 8h
Hawaii - Aleutian	Hawaii and a portion of the Aleutian Islands that is west of 169 degrees 30 minutes west longitude observes the Hawaii-Aleutian Standard Time Zone. Although Hawaii does not observe daylight saving time the Aleutian Islands do observe daylight saving time.	UTC - 10h	UTC - 9h Hawaii does not observe daylight saving time

Using the Timezone-Set Wizard

You can configure the timezone on the Oracle Enterprise Session Border Controller by running a timezone-set wizard from the root location via the ACLI. Use the following procedure to configure the Oracle Enterprise Session Border Controller timezone. If you need to exit the timezone-set command before completing it, use the key sequence Ctrl-D.

Note:

Note: The procedure described below may display different prompts depending on whether your system is running on VXWorks or LINUX.

To configure the timezone:

1. At the root prompt, enter timezone-set and press Enter.

```
ACMEPACKET# timezone-set
```

The following displays.

```
=========================================
Calling tzselect. Use ^D to cancel without save
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa
2) Americas
3) Antarctica
4) Arctic Ocean
5) Asia
6) Atlantic Ocean
7) Australia
8) Europe
9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#?
=========================================
```

2. Enter the number corresponding to the continent or ocean you want to select, and press Enter. Or enter none to specify the time zone using the Portable Operating System Interface (POSIX) timezone format.

Note: For a procedure to configure timezones using POSIX format, see *Configuring Timezone using POSIX Format*.

```
#? 2
```

The following displays.

```
=========================================
Please select a country.
1) Anguilla
2) Antigua & Barbuda
3) Argentina
4) Aruba
5) Bahamas
6) Barbados
7) Belize
8) Bolivia
9) Bonaire Sint Eustatius & Saba
10) Brazil
11) Canada
12) Cayman Islands
13) Chile
14) Colombia
15) Costa Rica
16) Cuba
17) Curacao
18) Dominica
19) Dominican Republic
20) Ecuador
21) El Salvador
22) French Guiana
23) Greenland
24) Grenada
25) Guadeloupe
26) Guatemala
27) Guyana
28) Haiti
29) Honduras
30) Jamaica
31) Martinique
32) Mexico
33) Montserrat
34) Nicaragua
```

```
35) Panama
36) Paraguay
37) Peru
38) Puerto Rico
39) Sint Maarten
40) St Barthelemy
41) St Kitts & Nevis
42) St Lucia
43) St Martin (French part)
44) St Pierre & Miquelon
45) St Vincent
46) Suriname
47) Trinidad & Tobago
48) Turks & Caicos Is
49) United States
50) Uruguay
51) Venezuela
52) Virgin Islands (UK)
53) Virgin Islands (US)
#?
========================================
```

3. Enter the number corresponding to the country you want to select, and press Enter.

```
#? 49
```

The following displays.

```
========================================
Please select one of the following time zone regions.
1) Eastern Time
2) Eastern Time - Michigan - most locations
3) Eastern Time - Kentucky - Louisville area
4) Eastern Time - Kentucky - Wayne County
5) Eastern Time - Indiana - most locations
6) Eastern Time - Indiana - Daviess, Dubois, Knox & Martin Counties
7) Eastern Time - Indiana - Pulaski County
8) Eastern Time - Indiana - Crawford County
9) Eastern Time - Indiana - Pike County
10) Eastern Time - Indiana - Switzerland County
11) Central Time
12) Central Time - Indiana - Perry County
13) Central Time - Indiana - Starke County
14) Central Time - Michigan - Dickinson, Gogebic, Iron & Menominee Counties
15) Central Time - North Dakota - Oliver County
16) Central Time - North Dakota - Morton County (except Mandan area)
17) Central Time - North Dakota - Mercer County
18) Mountain Time
19) Mountain Time - south Idaho & east Oregon
20) Mountain Time - Navajo
21) Mountain Standard Time - Arizona
22) Pacific Time
23) Alaska Time
24) Alaska Time - Alaska panhandle
25) Alaska Time - southeast Alaska panhandle
26) Alaska Time - Alaska panhandle neck
27) Alaska Time - west Alaska
28) Aleutian Islands
29) Metlakatla Time - Annette Island
30) Hawaii
#?
========================================
```

4. Enter the number corresponding to the time zone region you want to select, and press Enter.

```
#? 1
```

The following displays.

```
==========================================
The following information has been given:
        United States
        Eastern Time
Therefore TZ='America/New_York' will be used.
Local time is now:      Wed Mar 13 11:18:52 EDT 2013.
Universal Time is now:  Wed Mar 13 15:18:52 UTC 2013.
Is the above information OK?
1) Yes
2) No
#?
==========================================
```

5. Enter 1 (Yes) and press Enter. Or enter 2 (No) to go back to Step 2 and enter the correct timezone information.

```
#? 1
```

The following displays.

```
==========================================
Timezone=America/New_York
ACMEPACKET#
==========================================
```

You have completed the timezone-set wizard.

Configuring Timezone using POSIX Format

If you want to configure the timezone using POSIX format, you can select the option none - I want to specify the time zone using the Posix TZ format. in Step 2 of the timezone-set wizard.

Note: If you need to exit the timezone-set command before completing it, use the key sequence Ctrl-D.

To set the timezone using POSIX format:

1. At the root prompt, enter timezone-set and press Enter.

```
ACMEPACKET# timezone-set
```

The following displays.

```
==========================================
Calling tzselect. Use ^D to cancel without save
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa
2) Americas
3) Antarctica
4) Arctic Ocean
5) Asia
6) Atlantic Ocean
7) Australia
8) Europe
9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#?
==========================================
```

2. Enter 11, and press Enter.

```
#? 11
```

The following displays.

```
==========================================
Please enter the desired value of the TZ environment variable.
```

```
For example, GST-10 is a zone named GST that is 10 hours ahead (east) of
UTC.
==========================================
```

3. Enter the UTC/GMT value for your location. For valid UTC/GMT values, see the *Timezone Offsets Table (132)*.

```
#? UTC-10
```

The following displays.

```
==========================================
The following information has been given:
        TZ='UTC-10'
Therefore TZ='UTC-10' will be used.
Local time is now:      Thu Apr 11 02:50:18 UTC 2013.
Universal Time is now:  Wed Apr 10 16:50:18 UTC 2013.
Is the above information OK?
1) Yes
2) No
#?
==========================================
```

4. Enter 1 (Yes), and press Enter. Or enter 2 (No) to go back to Step 2 and enter the correct timezone information.

```
#? 1
```

The following displays. If you specified a value that does not relate to your Oracle Enterprise Session Border Controller location, a warning displays.

```
==========================================
Timezone=UTC-10
WARNING: custom timezone will apply to application only.
ACMEPACKET#
==========================================
```

You have completed the timezone-set wizard.

Tailored Configuration Views for SIPTX

In E-C[xz]6.4.0, tailored configuration views are available only for information related to SIP Trunk Xpress (SIPTX).

You can tailor the configuration information the system displays when you run the show running-config (abbreviated as show run) and show configuration commands by adding to them the generated, static, or both arguments.

- generated—Shows only template-generated configuration data, i.e., the result of putting templates into effect.

```
ACMEPACKET# show running generated
```

- static—Shows configuration you have performed directly on the system (without using templates) and that is saved on the system, and configuration data for instantiated templated services. This is the same as only specifying show run or show config.

```
ACMEPACKET# show running static
```

- both—Shows all configuration data.

```
ACMEPACKET# show running both
```

When you use the show running generated and show configuration generated, you can add one of several options for tailoring your displays. You can also send the output to a file you designate. These tailoring arguments correspond to the kinds of templates you can use:

- by-profile—Displays configuration generated by template profiles, or intermediary configurations containing subelements; each service you use must be associated with a profile, even if that profile is empty. The command requires you enter a profile name; only configurations generated by the specified profile are displayed.

```
ACMEPACKET# show running generated by-profile NewProfile
```

- by-service—Displays configuration generated by template services, or instantiations of a template. The command requires you enter a service name; only configurations generated by the specified service are displayed.

```
ACMEPACKET# show running generated by-service NewService
```

- by-template—Displays configuration data generated by a template. The command requires you enter a template name; only configurations generated by the specified template are displayed.

```
ACMEPACKET# show running generated by-template NewTemplate
```

- short—Displays only the attributes/values you have modified in the configuration.

```
ACMEPACKET# show running generated short
```

- to-file—Sends the output you display to the file you specify.

```
ACMEPACKET# show running generated to-file /ramdrv/logs showRunningGen
```

Showing Running for Templated Configuration

The output from the show run command looks different for configuration generated using templates. The only templated configuration you can view is related to SIP Trunk Xpress (SIPTX).

The following example shows what the output for show run looks like with configuration produced from a template. Only profiles, services, and statically-produced configuration are displayed.

```
ACMEPACKET# show running-config
profile
         profile-name defaultAccounting
         template-name Accounting
         profile-state                              enabled
         acctProto
         acctState
         acctGenStart
         acctGenInterim
         acctAddress
service
         service-name accounting1
         profile-name defaultAccounting
         service-state                              enabled
         acctProto                                  RADIUS
         acctState                                  enabled
         acctGenStart                               INVITE
         acctGenInterim                             OK
         acctAddress 172.16.11.2
```

The following example shows what the output for show run generated looks like when a configuration is generated by a template. The display shows whether the configuration was generated by a profile, service, or template.

```
ACMEPACKET# show running-config generated
account-config
         hostname                                   localhost
         port                                       1813
         strategy                                   Hunt
         protocol                                   RADIUS
         state                                      enabled
         max-msg-delay                              60
         max-wait-failover                          100
         trans-at-close                             disabled
         generate-start                             Invite
         generate-interim Reinvite-Response
                                                    OK
         intermediate-period                        0
         file-output                                disabled
         file-path /ramdrv/logs/
         max-file-size                              1000000
         max-files                                  5
         file-compression                           disabled
         file-rotate-time                           0
         options
         file-delete-alarm                          disabled
         ftp-push                                   disabled
```

```
        ftp-address
        ftp-port                                21
        ftp-user
        ftp-password
        ftp-remote-path
        cdr-output-redundancy                   enabled
        interim-stats-id-types
        account-servers
                hostname 172.16.11.2
                port 1813
                state enabled
                min-round-trip 250
                max-inactivity 60
                restart-delay 30
                bundle-vsa enabled
                secret
                NAS-ID
                priority                                0
                origin-realm
                domain-name-suffix
        prevent-duplicate-attrs                 disabled
        vsa-id-range
        cdr-output-inclusive                    disabled
        ftp-strategy                            Hunt
        ftp-max-wait-failover                   120
        generated-by-template Accounting
        generated-by-profile defaultAccounting
        generated-by-service accounting1
```

Tailoring Show Running for Templates

You can use the following commands to display information for a configuration made from a template. The only templated configuration that you can view is related to SIP Trunk Xpress (SIPTX).

- show run service—Displays all services you created using a template.
- show run prfile—Displays all profiles you created using a template.
- show run static-vars—Displays all status variables you created using a template.

Quit Command in Configuration Mode

When using the "quit" command after saving a configuration, the system exits the configuration mode and displays the root prompt for the Oracle Enterprise Session Border Controller.

In the following example, the “quit” command is used and the session-recording-server configuration is saved. The root prompt of ACMEPACKET displays.

```
ACMEPACKET(session-recording-server)# quit
Save Changes [y/n]?: y
session-recording-server
        name                                    crs2
        description
        realm                                   2.2.2.2
        mode                                    selective
        destination                             3.3.3.3
        port                                    5060
        transport-method                        DynamicTCP
        ping-method
        ping-interval                           0
        last-modified-by                        admin@10.1.25.17
        last-modified-date                      2013-06-06 11:49:24
ACMEPACKET#
```

Update the Configuration Schema

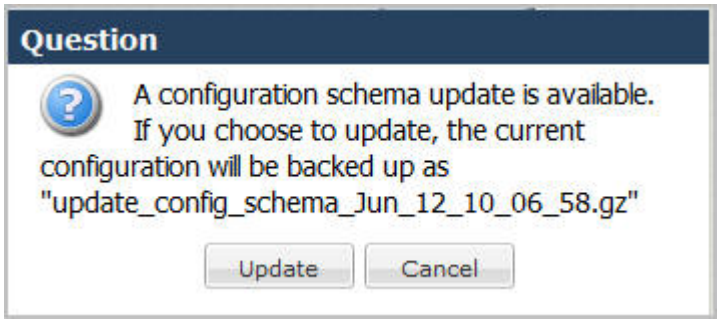
You can update the configuration parameters in your software with any new parameters included in a subsequent release by updating the schema.

Updating the schema adds any new parameters to each configuration screen in Basic Mode.

After updating your Web GUI software to a subsequent release, the system displays a schema update prompt after first log on to the GUI. If you click Cancel, the update is bypassed and no new parameters are added. The update prompt displays each time you log on to the Web GUI, until you choose to update the configuration schema.

Procedure

1. Log into the Web GUI. The system displays the following prompt.



2. Click **Update**. The system backs up the current configuration and updates the configuration schema.

   Note: If needed, you can reinstall the backed up configuration at a later time from the System tab in the Web GUI.

3. Click **OK**.
4. On the Configuration page toolbar, click **Save**.

Physical Interfaces: Acme Packet 3820 and Acme Packet 4500

There are two sets of physical interfaces on the network interface unit (NIU) used in the Acme Packet 3820 and Acme Packet 4500. These interfaces are located on the rear of the system chassis.

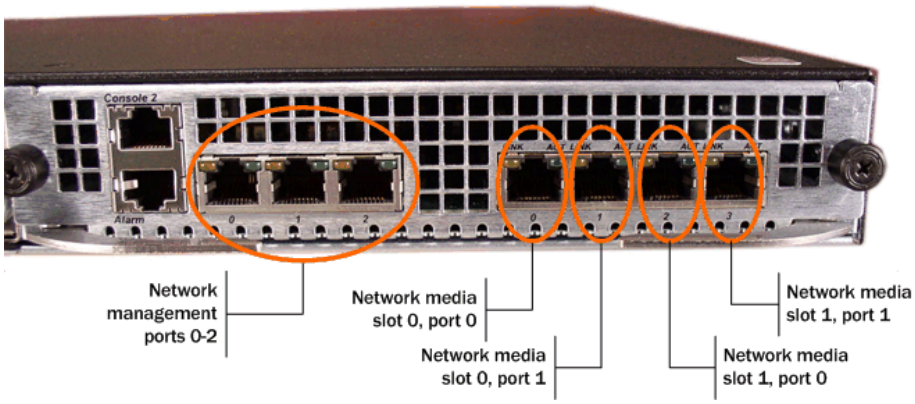
- Media interfaces are on the network interface unit (NIU); they are also referred to as network media ports
- Management interfaces are also on the NIU; they are also referred to as network management ports

The following picture of the NIU shows you how the network media and network management ports appear. These designations are an important point of reference when you set up physical interface configurations. Note that the slot parameter for network management ports will always be set to zero (0).

Network Media Interfaces

The NIU installed on your Acme Packet 3820 or Acme Packet 4500 determines the number of interfaces, hardware protocol, and connection speed available for media and signaling traffic.

- The NIU offers either four ports , and can use single mode or multimode fiber with an LC connector.
  - 4-port GigE copper (RJ45)
  - 4-port GigE SFP (LX, SX, or Copper)
  - 4-port GigE SFP with QoS and IPSec (LX, SX, or Copper)
  - 4-port GigE SFP with IPSec (LX, SX, or Copper)
  - 4-port GigE SFP with QoS (LX, SX, or Copper)
  - 4-port GigE SFP ETC NIU (LX, SX, or Copper)



Network Management Interfaces

The first management interface (labeled port 0 on the NIU's group of management ports) is used to carry traffic such as:

- SNMP
- Telnet
- SSH
- FTP
- ACP/XML
- Logs sent from the Oracle Enterprise Session Border Controller
- Boot the Oracle Enterprise Session Border Controller from a remote file server

The other two rear interfaces (port 1 and port 2) are used for state replication for high availability (HA). For HA, these interfaces on the Acme Packet 3820 and Acme Packet 4500 are directly connected by a crossover cable.

The following table summarizes the physical interface configuration parameters, which interface they are applicable to, and whether they are required.

Parameter	Network Media Interface	Network Management Interface
name	R	R
operation-type	R	R
port	R	R
slot	R	R
virtual-mac	O	I
admin-state	R	I
auto-negotiation	R	I
duplex-mode	R	I
speed	R	I
wancom-health-score	I	O
R = Required, O = Optional, I = Invalid		

Before You Configure

This section describes steps you should take prior to configuring physical interfaces.

Before you configure a physical interface:

1. Decide on the number and type of physical interfaces you need.

   For example, you might have one media interface connecting to a private network and one connecting to the public network. You might also need to configure maintenance interfaces for HA functionality.
2. Determine the slot and port numbering you will need to enter for the physical interfaces you want to configure. The graphic above can serve as your slot and port numbering reference.
3. If you are configuring your Acme Packet 4500 for HA, refer to the HA Nodes documentation and follow the instructions there for setting special parameters in the physical interface configuration.

Physical Interface Configuration

This section describes how to configure the name, location, and Ethernet parameters for Oracle Enterprise Session Border Controller physical interfaces.

To add a physical interface:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```
3. Type phy-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# phy-interface
   ACMEPACKET(phy-interface)#
   ```

   From this point, you can configure physical interface parameters. To view all physical interfaces parameters, enter a ? at the system prompt.

   The following is an example what an physical interface configuration might look like. Parameters not described in this section are omitted below.

   ```
   phy-interface
           name                            s0p0
           operation-type                  Media
           port                            0
           slot                            1
           admin-state                     enabled
           auto-negotiation                disabled
           duplex-mode                     FULL
           speed                           100
   ACMEPACKET(phy-interface)#
   ```

Identity and State

You must specify the identity for all types of physical interfaces, and the state for media interfaces.

Set the following parameters to configure the identity and state of a physical interface:

1. name—Set a name for the interface using any combination of characters entered without spaces. For example: Internet (for a Fast Ethernet media and signaling interface) or maint0 (for a maintenance interface).
2. admin-state—Leave the administrative state parameter set to enabled to receive and send media and signaling on an interface. Select disabled to prevent media and signaling from being received and sent. The default for this parameter is enabled. The valid values are:

   - enabled | disabled

Operation Type and Location

The following parameters determine the physical interface card and port you are about to configure.

Set the following parameters to configure the operation type and location for a physical interface:

1. operation-type—Select the type of physical interface connection to use. The default value is control. The valid values are:
   - media—Front-panel interfaces only. Port: 0-3; Slot: 0 or 1
   - maintenance—Rear-panel interface only. Port: 0, 1, or 2; Slot: 0
   - control—Rear-panel interfaces only. Port 0, 1, or 2; Slot: 0
2. slot—Select the physical slot number on the Oracle Enterprise Session Border Controller chassis. The default is 0. The valid values are:
   - 0 is the motherboard (rear-panel interface) if the name begins with wancom
   - 0 is the left two media Phy media slot on front of the Oracle Enterprise Session Border Controller Chassis
   - 1 is the right Phy media slot on front of the Oracle Enterprise Session Border Controller Chassis (front and rear interfaces)
3. port—Set the port. From left to right as you face the chassis, the possible values are:
   - 0-3—For four possible GigE ports on the front of the Oracle Enterprise Session Border Controller chassis
   - 0-3—For four possible FastE ports on the front of the Oracle Enterprise Session Border Controller chassis
   - 0-2—Rear interfaces

Auto-negotiation for Management Interfaces

For network management interfaces (wancom), set the parameters that enable or disable auto-negotiation, duplex mode, and the data rate.

Set the following parameters to configure auto-negotiation for management interfaces:

1. **auto-negotiation**—Leave this parameter set to enabled so that the Oracle Enterprise Session Border Controller and the device to which it is linked can automatically negotiate the duplex mode and speed for the link.

   If auto-negotiation is enabled, the Oracle Enterprise Session Border Controller begins to negotiate the link to the connected device at the duplex mode you configure. If auto-negotiation is disabled, then the Oracle Enterprise Session Border Controller will not engage in a negotiation of the link and will operate only at the duplex mode and speed you set. The default is enabled. The valid values are:
   - enabled | disabled
2. **duplex-mode**—Set the duplex mode. The default is full.

   Given an operating speed of 100 Mbps, full duplex mode lets both devices on a link send and receive packets simultaneously using a total bandwidth of 200 Mbps. Given the same operating speed, half duplex mode limits the devices to one channel with a total bandwidth of 100 Mbps. The valid values are:
   - half | full
3. **speed**—Set the speed in Mbps of the management physical interfaces; this field is only used if the auto-negotiation field is set to disabled. 100 is the default. The valid values are:
   - 10 | 100 | 1000

HA Configuration

Refer to this guide's *HA Nodes documentation* for more information about when and how to configure virtual-mac and wancom-health-score parameters. If you are not using HA, you can leave these parameters set to their defaults.

Interface Utilization Graceful Call Control Monitoring and Fault Management

When you enable this feature, the Oracle Enterprise Session Border Controller monitors network utilization of its media interfaces and sends alarms when configured thresholds are exceeded. You can also enable overload protection on a per-media interface basis, where the Oracle Enterprise Session Border Controller will prevent call initializations during high traffic but still allow established calls to continue if traffic passes the critical threshold you define.

Calculation Overview

When enabled to do so, the Oracle Enterprise Session Border Controller performs a network utilization calculation for each of its media ports. This calculation takes into account rates of receiving and transmitting data, the speed at which each is taking place, and the quality of data traversing the interface. The Oracle Enterprise Session Border Controller keeps statistics for each media port so it can compare previously- and newly-retrieved data. For heightened accuracy, calculations are performed with milliseconds (rather than with seconds).

Alarms

In the physical interface configuration, you can establish up to three alarms per media interface—one each for minor, major, and critical alarm severities. These alarms do not have an impact on your system's health score. You set the threshold for an alarm as a percentage used for receiving and transmitting data.

For example, you might configure the following alarms:

- Minor, set to 50%
- Major, set to 70%
- Critical, Set to 90%

When the utilization percentage hits 50%, the system generates a minor alarm. At 70%, the system clears the minor alarm and issues a major one. And at 90%, the system clears the major alarm and issues a critical one. At that point, if you have overload protection enabled, the system will drop call initiations but allow in-progress calls to complete normally.

To prevent alarm thrashing, utilization must remain under the current alarm threshold for 10 seconds before the system clears the alarm and rechecks the state.

Alarm Configuration

This section shows you how to configure alarm thresholds and overload protection per media interface.

Configuring Utilization Thresholds for Media Interfaces

To configure utilization thresholds for media interfaces:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type phy-interface and press Enter. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

```
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)#
```

4. Type network-alarm-threshold and press Enter.

```
ACMEPACKET(phy-interface)# network-alarm-threshold
ACMEPACKET(network-alarm-threshold)#
```

5. severity—Enter the severity for the alarm you want to fine for this interface: minor (default), major, or critical. Since the parameter defaults to minor, you must change the value if you want to define a major or critical alarm.
6. value—Enter the percentage of utilization (transmitting and receiving) for this interface that you want to trigger the alarm. For example, you might define a minor alarm with a utilization percentage of 50. Valid values are between 0 and 100, where 0 is the default.
7. Save your work.

Configuring Graceful Call Control

You can enable the Oracle Enterprise Session Border Controller to stop receiving session-initiating traffic on a media interface when the traffic for the interface exceeds the critical threshold you define for it.

To enable graceful call control:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type phy-interface and press Enter. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

```
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)#
```

4. overload-protection—Change this parameter's value to enabled if you want to turn graceful call control on. Leave it set to disabled (default) if you do not want to use this feature.
5. Save your work.

Network Interfaces

The network interface element specifies a logical network interface. In order to use a network port on a network interface, you must configure both the physical interface and the corresponding network interface configuration elements. If the network interface does not use VLANs tagging, ensure that the **sub-port-id** parameter is set to 0, the default value. When VLAN tags are used on a network interface, the valid **sub-port-id** value can range from 1-4096. The combination of the **name** parameter and the **sub-port-id** parameter must be unique in order to identify a discrete network interface.

IP Configuration

A Oracle Enterprise Session Border Controller network interface has standard parameters common to nearly all IP network interfaces. There are a few fields that are unique to the Oracle Enterprise Session Border Controller.

VLANs

VLANs are used to logically separate a single physical interface into multiple network interfaces. There are several applications for this like MPLS VPNs (RFC 2547), MPLS LSPs, L2VPNs (IPSec, L2TP, ATM PVCs), reusing address space, segmenting traffic, and maximizing the bandwidth into a switch or router. The range of services and management capabilities you can implement with VPNs is huge.

The primary applications of VLANs on the Oracle Enterprise Session Border Controller are VPNs and peering. Several peering partners may terminate their connections to a Oracle Enterprise Session Border Controller on a single physical interface. VLAN tags are used to segregate and correctly route the terminated traffic. The Oracle Enterprise Session Border Controller can support a maximum of 1024 VLANs per physical interface. Ingress packets that do not contain the correct VLAN tag will be dropped. All packets exiting on an egress interface will have the VLAN tag appended to them.

The Oracle Enterprise Session Border Controller can be included in an MPLS network through its connectivity to a PE router, which maps a MPLS VPN label to an 802.1q VLAN tag. Each Oracle Enterprise Session Border Controller can terminate different 802.1q VLANs into separate network interfaces, each of which can represent a different customer VPN.

Overlapping Networks

Overlapping networks are when two or more private networks with the same addressing schemes terminate on one physical interface. The problem this creates can easily be solved by using VLAN tagging. For example, two 10.x.x.x networks terminating on one Oracle Enterprise Session Border Controller network interface will obviously not work. The Oracle Enterprise Session Border Controller includes the IP Address, Subnet Mask, and 802.1q VLAN tag in its Network Interface determination. This allows Oracle Enterprise Session Border Controller to directly interface to multiple VPNs with overlapping address space.

Administrative Applications Over Media Interfaces

By default, the Oracle Enterprise Session Border Controller's FTP, ICMP, SNMP, SSH, and Telnet services cannot be accessed via the media interfaces. In order to enable these services, you must explicitly configure access by identifying valid source addresses for the specific applications. Doing such uses the Oracle Enterprise Session Border Controller's host-in-path (HIP) functionality.

When traffic is received on media interfaces, it is scanned for FTP, ICMP, SNMP, SSH, or Telnet packets. The configuration is set to identify the possible IP addresses where that traffic may be sourced from. When a match is made among packet type and source address, those packets are forwarded through the media interfaces to the processes running on the system's CPU.

Each media **network-interface**'s gateway should be configured so that off-subnet return traffic can be forwarded out the appropriate media interface. Also, it is advisable that no overlapping networks are configured between any media network interface and the administrative interfaces (wancom).

Configurable MTU Size

Configurable MTU on per network-interface basis enables the user to set a different MTU on each network interface. It also enables the user to set a system wide default MTU for IPv6 and IPv4 network interfaces. System wide defaults can be set in **system-config** configuration object by setting **ipv6-signaling-mtu** or **ipv4-signaling-mtu**. Defaults are 1500 for both IPv6 and IPv4.

These settings can be overwritten for each network interface by setting **signaling-mtu** in **network-interface** configuration object. Default is 0 – meaning use the system wide MTU.

This feature applies to all Signaling packets generated by the Oracle Enterprise Session Border Controller. All UDP packets greater than the MTU will be fragmented. For all TCP connections we advertise MSS (Maximum Segment Size) TCP option in accordance with the configured MTU. MSS option is sent in SYN and SYN/ACK packets to let the other side of the TCP connection know what your maximum segment size is. This ensures that no TCP packet is greater than the configured MTU.

1. MTU settings do not apply to media packets.
2. UDP: MTU settings apply only to packets sent by the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller will continue to process received packets even if they exceed to the configured MTU.
3. Security Phy (IPsec) hardware only; We subtract 100 bytes from the configured MTU to allow for extra headers added by security protocols. This happens even when Security Phy (IPsec) is in clear mode (no security is being applied). Due to hardware limitations of the Security Phy (IPsec) it only allows one MTU per physical port. The maximum MTU of all network interfaces on a given physical port will be used as the MTU for that physical port.
4. The Call Recording feature is where we make a copy of a packet, encapsulate it in an IP-in-IP header and send it to a configured Call Recording Server (CRS). When Call Recording is enabled, to allow space for IP-in-IP encapsulation we reduce the MTU of the original packets to be to be the lesser of the two options listed below.
   - Original Destination network MTU minus size of IP-in-IP header.
   - CRS network interface's MTU minus size of IP-in-IP header.

   Note: This will ensure that the traffic sent to the CRS will be within the MTU constraints of CRS' network-interface.

Network Interface Configuration

This section explains how to access and configure network interface.

Special Considerations

Configuration changes to network interface parameters might have an impact on boot configuration parameters. After configuring the network interface, you might receive a message indicating that you could be changing boot config parameters under the following circumstances:

- A physical interface or network interface element matches the boot interface (for example, the physical port is the same as the boot port).
- The boot configuration parameters are modified, because the IPv4 address, netmask, or gateway is different from the corresponding boot configuration parameters.

You are asked if you want to continue. If you enter yes, the configuration will be saved and then the differing boot configuration parameters will be changed. If you enter no, then the configuration is not saved and the boot configuration parameters are not changed.

Configuring the physical and network interface elements for the first management interface is optional because that interface, eth0, is implicitly created by a valid bootparam configuration that specifies the boot device, IPv4 address, subnet, and gateway.

Network Interfaces Configuration

This section describes how to configure a network interface.

Access the **network-interface** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)
```

IP Configuration and Identification

You must specify the identity and address for all network interfaces.

Set the following parameters to configure a network interface:

1. **name**—Set the name for the network interface. This must be the same name as the physical interface to which it corresponds.
2. **description**—Enter a description of the network for easier identification.
3. **hostname**—Set the hostname (FQDN) of this network interface. This parameter is optional.
4. **ip-address**—Set the IP address of this network interface.
5. **netmask**—Set the netmask of this network interface in dotted decimal notation.
6. **gateway**—Set the gateway that this network interface uses to communicate with the next hop.
7. **sec-gateway**—Set an additional optional gateway for this network interface
8. **dns-ip-primary**—Set the DNS servers. You can set an additional two DNS servers by using the **dns-ip-backup1** and **dns-ip-backup2** parameters.
9. **dns-domain**—Set the default domain name.
10. **signaling-mtu**—Sets the MTU size for IPv4 or IPv6 transmission.

VLAN Configuration

One parameter is required to configure VLANs on a Oracle Enterprise Session Border Controller. The **sub-port-id** parameter located in the **network-interfaces** element adds and masks for a specific VLAN tag.

sub-port-id—Enter the identification of a specific virtual interface in a physical interface (e.g., a VLAN tab). If this network interface is not channelized, leave this field blank, and the value will correctly default to 0. The **sub-port-id** is only required if the operation type is Media. The valid range is:

- Minimum—0
- Maximum—4095.

HIP Address Configuration

To configure administrative service functionality on a media interface, you must first define all source IP addresses in the media-interface's network that will exchange administrative traffic with the system. Next you will identify the type of administrative traffic each of those addresses will exchange.

You must configure the **gateway** parameter on this **network-interface** for administrative traffic to successfully be forwarded. You should also ensure that this network interface is not on an overlapping network as any of the administrative networks (wancoms).

Set the following parameters to configure HIP functionality on a network interface:

1. **add-hip-ip**—Configure all possible IP address(es) from which the Oracle Enterprise Session Border Controller will accept administrative traffic. Entries in this element are IP addresses of media network interfaces. This parameter can accept multiple IP addresses. You can later remove this entry by typing **remove-hip-ip** followed by the appropriate IP address.
2. **add-ftp-ip**—Set the IP address(es) that will access the Oracle Enterprise Session Border Controller's FTP server. This allows standard FTP packets enter the Oracle Enterprise Session Border Controller and reach the host. You can later remove this entry by typing **remove-ftp-ip** followed by the appropriate IP address.
3. **add-icmp-ip**—Set the IP address(es) that can ping the system and expect replies. This parameter can accommodate multiple ping IP addresses. You can later remove this entry by typing **remove-icmp-ip** followed by the appropriate IP address.

   For security, if the ICMP address and the hip-ip-list are not added for an address, the Oracle Enterprise Session Border Controller hardware discards ICMP requests or responses for the address.
4. **add-snmp-ip**—Set the IP address(es) that will access the system's SNMP process. This lets SNMP traffic enter the Oracle Enterprise Session Border Controller and reach the host. You can later remove this entry by typing **remove-snmp-ip** followed by the appropriate IP address.
5. **add-telnet-ip**—Set the IP address(es) that can connect and access the system through Telnet. You can later remove this entry by typing **remove-telnet-ip** followed by the appropriate IP address.
6. **add-ssh-ip**—Set the IP address(es) that can connect and access the system through SSH. You can later remove this entry by typing **remove-SSH-ip** followed by the appropriate IP address.

Configurable MTU Size

Configurable MTU on per network-interface basis enables the user to set a different MTU on each network interface. It also enables the user to set a system wide default MTU for IPv6 and IPv4 network interfaces. System wide defaults can be set in **system-config** configuration object by setting **ipv6-signaling-mtu** or **ipv4-signaling-mtu**. Defaults are 1500 for both IPv6 and IPv4.

These settings can be overwritten for each network interface by setting **signaling-mtu** in **network-interface** configuration object. Default is 0 – meaning use the system wide MTU.

This feature applies to all Signaling packets generated by the Oracle Enterprise Session Border Controller. All UDP packets greater than the MTU will be fragmented. For all TCP connections we advertise MSS (Maximum Segment Size) TCP option in accordance with the configured MTU. MSS option is sent in SYN and SYN/ACK packets to let the other side of the TCP connection know what your maximum segment size is. This ensures that no TCP packet is greater than the configured MTU.

1. MTU settings do not apply to media packets.
2. UDP: MTU settings apply only to packets sent by the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller will continue to process received packets even if they exceed to the configured MTU.
3. Security Phy (IPsec) hardware only; We subtract 100 bytes from the configured MTU to allow for extra headers added by security protocols. This happens even when Security Phy (IPsec) is in clear mode (no security is being applied). Due to hardware limitations of the Security Phy (IPsec) it only allows one MTU per physical port. The maximum MTU of all network interfaces on a given physical port will be used as the MTU for that physical port.

4. The Call Recording feature is where we make a copy of a packet, encapsulate it in an IP-in-IP header and send it to a configured Call Recording Server (CRS). When Call Recording is enabled, to allow space for IP-in-IP encapsulation we reduce the MTU of the original packets to be to be the lesser of the two options listed below.
   - Original Destination network MTU minus size of IP-in-IP header.
   - CRS network interface's MTU minus size of IP-in-IP header.

   Note: This will ensure that the traffic sent to the CRS will be within the MTU constraints of CRS' network-interface.

System Wide MTU Size

To change system wide MTU settings:

1. Access the **system-config** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

2. Type **select** to begin editing the **system-config** object.

```
ACMEPACKET(system-config)# select
ACMEPACKET(system-config)#
```

3. **ipv6-signaling-mtu** or **ipv4-signaling-mtu** Configure MTU in the system config and optionally in the network interface. Default will be 1500 bytes.

```
ACMEPACKET(system-config)# ipv6-signaling-mtu 1500
ACMEPACKET(system-config)# ipv4-signaling-mtu 1600
```

4. Type **done** to save your configuration.

Session Border Controller (SBC) Deployment Behind a Network Address Translation (NAT) Device

The S-C[xz]6.3.9M4 release provides the *Support for SBC Behind NAT* SPL plug-in for deploying the Oracle Enterprise Session Border Controller on the private network side of a NAT device. The *Support for SBC Behind NAT* SPL plug-in changes information in SIP messages to hide the end point located inside the private network. The specific information that the *Support for SBC Behind NAT* SPL plug-in changes depends on the direction of the call, for example, from the NAT device to the SBC or from the SBC to the NAT device.

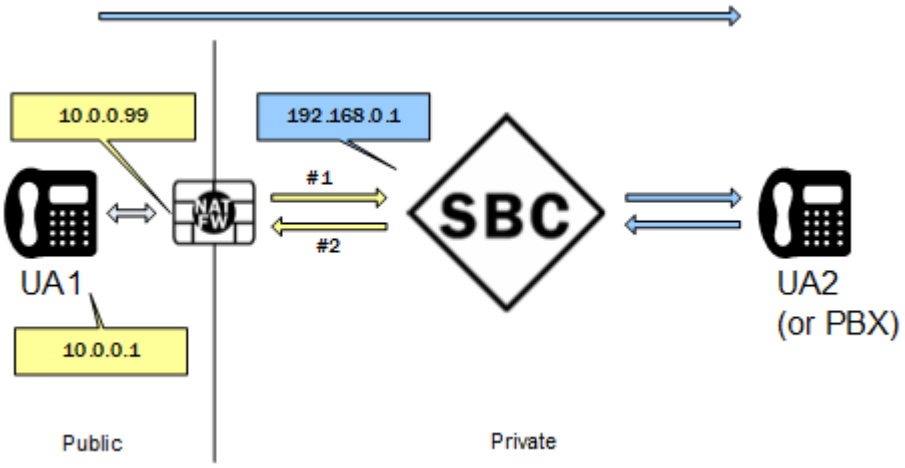
Configure the *Support for SBC Behind NAT* SPL plug-in for each SIP interface that is connected to a NAT device. One public-private address pair is required for each SIP interface that uses the SPL plug-in, as follows.

- The private IP address must be the same as the SIP Interface IP address.
- The public IP address must be the public IP address of the NAT device.

The following illustrations show the SBC deployed in the private network behind a NAT device, using the *Support for SBC Behind NAT* SPL plug-in. Examples follow each illustration to show where the *Support for SBC Behind NAT* SPL plug-in changes the SIP message information.

Call Initiated on the Access Side

In this illustration, UA1 invites UA2 to a session and UA2 responds.



#1. UA1 sends an INVITE through the NAT device to the Oracle Enterprise Session Border Controller with the following message.

```
INVITE sip:service@10.0.0.99:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.1:5060;branch=z9hG4bK-3539-1-0
Contact: sip:sipp@10.0.0.1:5060
…
Content-Type: application/sdp

o=user1 53655765 2353687637 IN IP4 10.0.0.1
c=IN IP4 10.0.0.1
…
```

The *Support for SBC Behind NAT* SPL plug-in looks for the public SIP Interface IP address 10.0.0.99 in R-URI, Via, Contact, and SDP. The SPL plug-in finds 10.0.0.99 in R-URI and changes it to the private SIP Interface IP address 192.168.0.1.

```
INVITE sip:service@192.168.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.1:5060;branch=z9hG4bK-3539-1-0
Contact: sip:sipp@10.0.0.1:5060
…
Content-Type: application/sdp

o=user1 53655765 2353687637 IN IP4 10.0.0.1
c=IN IP4 10.0.0.1
…
```

#2. The Oracle Enterprise Session Border Controller sends a Reply to UA1.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.0.0.1:5060;received=192.168.0.70;branch=z9hG4bK-3539-1-0
Contact: <sip:192.168.0.1:5060;transport=udp>
Content-Type: application/sdp
…
o=user1 53655765 2353687637 IN IP4 192.168.0.1
c=IN IP4 192.168.0.1
…
```

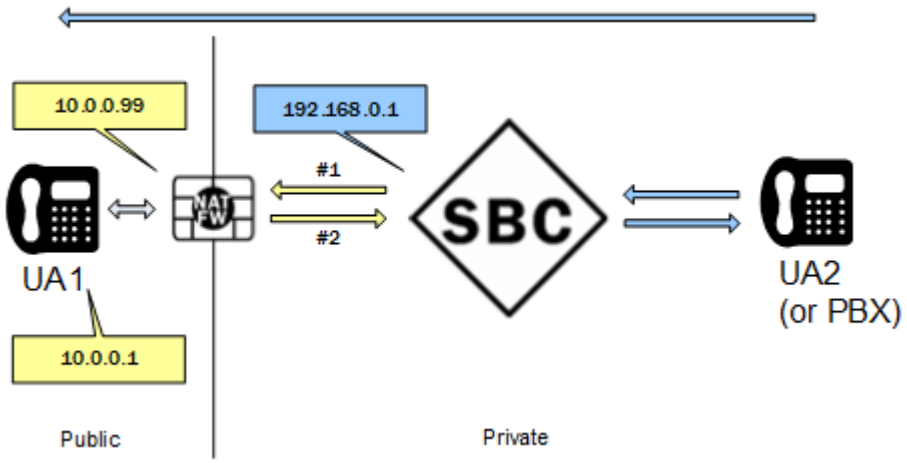
The *Support for SBC Behind NAT* SPL plug-in looks for the private SIP interface IP address 192.168.0.1 in R-URI, Via, Contact, and SDP. The SPL plug-in finds 192.168.0.1 in Contact and SDP and changes it to the public IP 10.0.0.99.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.0.0.1:5060;received=192.168.0.70;branch=z9hG4bK-3539-1-0
Contact: <sip:10.0.0.99:5060;transport=udp>
Content-Type: application/sdp
…
o=user1 53655765 2353687637 IN IP4 10.0.0.99
```

```
c=IN IP4 10.0.0.99
…
```

Call Initiated on the Core Side

In this illustration, UA2 invites UA1 to a session and UA1 responds.



#1. The Oracle Enterprise Session Border Controller sends an Invite to UA1.

```
INVITE sip:service@10.0.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bKbgs21h30a8kh8okcv790.1
Contact: <sip:sipp@192.168.0.1:5060;transport=udp>
Content-Type: application/sdp
…
o=user1 53655765 2353687637 IN IP4 192.168.0.1
c=IN IP4 192.168.0.1
…
```

The *Support for SBC Behind NAT* SPL plug-in looks for the private IP address 192.168.0.1 in R-URI, Via, Contact, and SDP. The SPL plug-in finds 192.168.0.1 in Via, Contact, and SDP and changes it to the public IP address 10.0.0.99.

```
INVITE sip:service@10.0.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.99:5060;branch=z9hG4bKbgs21h30a8kh8okcv790.1
Contact: <sip:sipp@10.0.0.99:5060;transport=udp>
Content-Type: application/sdp
…
o=user1 53655765 2353687637 IN IP4 10.0.0.99
c=IN IP4 10.0.0.99
…
```

#2. UA1 sends a Reply to the Oracle Enterprise Session Border Controller.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.0.0.99:5060;branch=z9hG4bKbgs21h30a8kh8okcv790.1
Contact: <sip: 10.0.0.1:5060;transport=UDP>
Content-Type: application/sdp
…
o=user1 53655765 2353687637 IN IP4 10.0.0.1
c=IN IP4 10.0.0.1
…
```

The *Support for SBC Behind NAT* SPL plug-in looks for the private SIP interface IP address 192.168.0.1 in R-URI, Via, Contact, and SDP. The SPL plug-in finds 192.168.0.1 in Via, changes it to the public IP 10.0.0.99.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bKbgs21h30a8kh8okcv790.1
Contact: <sip: 10.0.0.1:5060;transport=UDP>
Content-Type: application/sdp
```

```
…
o=user1 53655765 2353687637 IN IP4 10.0.0.1
c=IN IP4 10.0.0.1
…
```

Enable an SPL Plug-in

You must enable the SBC Programming Language (SPL) plug-in before you configure an SPL option. The process to enable the SPL plug-in on the Oracle Enterprise Session Border Controller (E-SBC) requires the following steps.

1. Upload the SPL plug-in to the E-SBC.
2. Add the SPL plug-in to the E-SBC configuration.
3. Execute the SPL file.
4. Synchronize the SPL files across HA pairs.
5. Reset the SPL on standby nodes.
6. Configure the SPL plug-in option.

1. Upload the SPL Plug-in to the E-SBC

Use any CLI or interface-based FTP or SFTP application to send the SPL plug-in to the E-SBC. You can use the wancom or eth0 management physical interface to reach the FTP/SFTP server on the SBC.

Upload the SPL plug-in to the /code/spl directory on the E-SBC.

2. Add the SPL Plug-in to the E-SBC Configuration

Confirm that you are in Superuser mode.

On the SBC, in the spl-configuration element, configure the SPL plug-in.

1. Type configure terminal, and press <Enter>.
2. Type system, and press <Enter>.
3. Type spl-config, and press <Enter>.
4. Type select, and press <Enter>.
5. Type plugins, and press <Enter>.
6. Type name, enter a space, and type the name of the SPL plug-in file.
7. Type done.
8. Type exit.
9. Type done.

3. Execute the SPL File

Confirm that the SPL plug-in is configured on the E-SBC, and that you exited the configuration menu.

You must save and activate the configuration.

Perform the save-config and activate-config operations on the E-SBC.

4. Synchronize the SPL Files Across HA Pairs

Confirm that you are in Superuser mode.

When running in an HA pair configuration, the active system and the standby system must both have the same version of the SPL plug-in installed. To facilitate configuring the standby system, you can execute the synchronize the spl ACLI command, without any arguments, to copy all of the files in the /code/spl directory from the active system to the same directory on the standby system. Note that any file on the standby system with the same name as a file on the active system is overwritten.

By adding the specific file name as an argument to the synchronize spl command, the individual, specified scripts are copied. For example:

ACMEPACKET# synchronize spl <name of the SPL plug-in file>

The synchronize spl command can only by executed from the active system in an HA pair. There is no means to synchronize spl files automatically during save and activate operations on the E-SBC.

Type synchronize spl, and press <Enter>.

5. Reset the SPL on Standby HA Nodes

Confirm that you are in Superuser mode.

Execute the reset spl command on all standby nodes that receive the spl file by way of the synchronize command.

Type reset spl, and press <Enter>.

6. Configure the SPL Plug-in Option

See the instructions for configuring the particular SPL plug-in option.

Configure the Session Border Controller (SBC) Behind a Network Address Translation (NAT) Device Option

Configure one public-private address pair for each SIP interface that uses the *Support for SBC Behind NAT* SPL plug-in, as follows.

- The private IP address must be the same as the SIP interface IP address.
- The public IP address must be the public IP address of the NAT device.

Before You Begin

- Confirm that the SIP interface is configured.
- Confirm that you are in the Superuser mode.

To configure the SIP interface IP addresses:

1. Type configure terminal, and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type system, and press <Enter>.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type session-router, and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

4. Type sip-interface, and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

5. Select the SIP interface, and press <Enter>.

```
ACMEPACKET(sip-interface)# select
<RealmID>:
1: DefaultENT 172.16.1.100:5060
2: DefaultSP  192.168.0.1:5060

selection:2
System_Primary(sip-interface)#
```

6. Type spl-options +HeaderNatPrivateSipIfIp "<value>", where <value> is the private SIP interface IP address, and press <Enter>.

```
ACMEPACKET(sip-interface)#spl-options +HeaderNatPrivateSipIfIp=192.168.0.1
```

7. Type spl-options +HeaderNatPublicSipIfIp "<value>", where <value> is the public IP address of the NAT device, and press <Enter>.

```
ACMEPACKET(sip-interface)#spl-options +HeaderNatPublicSipIfIp=10.0.0.99
```

8. Type done, and press <Enter>.
9. Save and activate the configuration.

Traceroute Command

The system can trace the route of an IP packet to an Internet host by sending probe packets and listening to responses from gateways along the route. Use the traceroute command to see each host route and the round trip time of packets received from each host in a route for diagnostic purposes.

The traceroute command sends probe packets that start with a maximum time-to-live (TTL) value of one. The system listens for an Internet Control Message Protocol (ICMP) error message in response to the TTL expiry, and records the source that sent the ICMP error message. The system repeats this process and increments the TTL value by 1 for each hop in the route to the final destination.

The traceroute command returns the following information, which allows tracing the packet route to its destination.

- TTL value
- IP address of each host along the route
- Amount of time that it takes for each probe packet to travel to each host in the route

Notes:

- Unless otherwise specified, the system sends three probe packets to each host.
- The traceroute command is only available in software versions of the Oracle Enterprise Session Border Controller, for example, Server Edition (SE) and Virtual Machine Edition (VME). For more information on supported platforms, see "Platform Support."

For traceroute command syntax and arguments, see "Traceroute Command Specifications."

Examples

The following example traces the route to IP address 172.30.0.167, identifying each host in the route and the amount of time that it takes for each of three probe packets to travel to each host. The first three probe packets reach the host at 172.44.0.1 in times ranging from less than one to a little over two milliseconds. The next three probe packets reach the route destination at IP address 172.30.0.167 all in less than one millisecond.

```
ACMEPACKET# traceroute 172.30.0.167
traceroute to 172.30.0.167
1 172.44.0.1 (0.669003 ms) (2.140045 ms) (2.290964 ms)
2 172.30.0.167 (0.25602 ms) (0.219822 ms) (0.604868 ms)
```

The following example traces the route to IP address 172.30.0.167 but specifies the use of 4 probe packets instead of the default of 3.

```
ACMEPACKET traceroute 172.30.0.167 probes 4
traceroute to 172.30.0.167
1 172.44.0.1 (0.549003 ms) (1.180045 ms) (2.920584 ms) (2.48541 ms)
2 172.30.0.167 (0.25802 ms) (0.220822 ms) (0.454868 ms) (0.387574)
```

The following example specifies that the traceroute command is issued to the IP address over the user-specified network interface private and VLAN 123.

```
ACMEPACKET traceroute 10.1.2.6 intf-name:vlan private:123
traceroute to 10.1.2.6
1 10.1.2.6 (0.265121 ms) (0.599080 ms) (0.0184195 ms)
```

The following example specifies that the wait for a response timeout is 4 seconds. The default value is three seconds.

```
ACMEPACKET traceroute 10.1.2.6 timeout 4
traceroute to 10.1.2.6
1 10.1.2.6 (0.265121 ms) (0.199080 ms) (0.0284195 ms)
```

The following example specifies that the traceroute starts at a user-specified source IP address of 172.20.22.31 to a destination IP address of 10.25.2.10.

```
ACMEPACKET traceroute 172.20.22.31 source-ip 10.25.2.10
traceroute to 172.20.22.31
172.20.22.31 (0.284121 ms) (0.499770 ms) (0.084595 ms)
```

SNMP

This section explains how to configure Simple Network Management Protocol (SNMP), trap receivers, and syslog servers. These features are not essential for baseline Oracle Enterprise Session Border Controller service, but they are necessary to use network management systems to manage the Oracle Enterprise Session Border Controller. They provide important monitoring and system health information that contribute to a robust deployment of the system.

Overview

SNMP is used to support monitoring of network-attached devices for conditions that warrant administrative attention. SNMP is comprised of three groups of settings on a Oracle Enterprise Session Border Controller. These settings are system-wide configurations including MIB contact information, SNMP community settings, and trap receivers.

Basic SNMP Parameters

The Oracle Enterprise Session Border Controller includes several parameters that control basic SNMP functionality. The MIB-related elements are for informational purposes, and are helpful if set. The remainder of the parameters determines if certain Oracle Enterprise Session Border Controller events are reported to the SNMP system.

SNMP Community

An SNMP community is a grouping of network devices and management stations used to define where information is sent and accepted. An SNMP device or agent might belong to more than one SNMP community. SNMP communities provide a type of password protection for viewing and setting management information within a community.

SNMP communities also include access level settings. They are used to define the access rights associated with a specific SNMP community. The Oracle Enterprise Session Border Controller lets you define two types of access levels: read-only and read-write. You can define multiple SNMP communities on a Oracle Enterprise Session Border Controller to segregate access modes per community and NMS host.

SNMP Trap Receiver Uses

A trap receiver is an application used to receive, log, and view SNMP traps for monitoring the Oracle Enterprise Session Border Controller (E-SBC). An SNMP trap is the notification sent from a network device, such as the E-SBC, that declares a change in service. Multiple trap receivers can be defined on an E-SBC for either redundancy or to segregate alarms with different severity levels to individual trap receivers.

Each Oracle Communications Session Delivery Manager which manages Oracle Enterprise Session Border Controllers should be configured on those SBCs as trap receivers.

SNMPv3 Support

The Oracle Enterprise Session Border Controller (E-SBC) supports SNMPv3, which provides the SNMP agent and SNMP Network Management System (NMS) with authentication, privacy, and access control during the delivery of secured traps. Currently, SNMPv3 traps are supported on the Net-Net ESD; SNMPv3 Get/Get-Bulk/Set actions are not supported at this time.

By default, the E-SBC supports SNMPv1v2. If you want to retain the existing SNMPv1v2 behavior, you do not need to update configuration. You can enable SNMPv3 at any time, at which point SNMPv1v2 configurations are ignored, and only SNMPv3 encrypted traps are sent to the associated external SNMP managers. Snmp-agent-mode, an attribute under system-config, allows you to select the mode that you want.

Configuring SNMP

This section describes how to configure your Oracle Enterprise Session Border Controller to work with external SNMP systems. Sample configurations are also provided.

SNMP Configuration Overview

1. Configure the SNMP identification information. This step includes configuring the MIB system contact, name, and location parameters.
2. Set the general SNMP parameters to enable or disable SNMP on the Oracle Enterprise Session Border Controller. This step includes setting the switches that govern how the SNMP system responds to specified events.
3. Set the syslog events. This step includes setting the parameters for handling SNMP monitoring syslog events, which can trigger SNMP syslog traps.
4. Set SNMP communities. Configuration is separated into a unique configuration element.
5. Set trap receivers. Configuration is separated into a unique configuration element.

SNMP Configuration

To configure SNMP:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type system-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

From this point, you can set SNMP parameters. The following is an example what an SNMP configuration might look like. Parameters not described in this section are omitted below.

```
system-config
        mib-system-contact             John Doe
        mib-system-name                Test System
        mib-system-location            Upstairs
        snmp-enabled                   enabled
        enable-snmp-auth-traps         disabled
        enable-snmp-syslog-notify      disabled
        enable-snmp-monitor-traps      disabled
        enable-env-monitor-traps       disabled
        snmp-syslog-his-table-length   1
        snmp-syslog-level              WARNING
```

System Wide Configuration for SNMP

This section describes the system-wide SNMP parameters found in the System Configuration element. These parameters set global SNMP information.

Set the following parameters to configure system wide SNMP functionality:

1. **mib-system-contact**—Set the contact information used within the system's MIB transactions. The SNMP agent sends this information to an NMS in response to an SNMP Get for the MIB-II sysContact MIB variable. This parameter's value can be a textual identification of your company's contact person for the system and information about how to contact that person.
2. **mib-system-name**—Set the identification of this Oracle Enterprise Session Border Controller presented within MIB transactions. This value, along with the target name of the system (identified in the boot parameters) are the

values reported for MIB-II when an SNMP GET is issued by the NMS for the MIB-II sysName variable. This parameter has no direct relation to the hostname parameter in the system configuration element.

By convention, this is the node's FQDN. For SNMP MIB-II sysName GETs, the system returns SNMP communications in the following format: <targetName>[.<mib-system-name>]

targetName is the value configured in the target name (tn) boot parameter and mib-system-name is the value configured in this field.

3. **mib-system-location**—Set the physical location of this Oracle Enterprise Session Border Controller that is reported within MIB transactions. This parameter is reported when an SNMP GET is issued by the NMS for the MIB-II sysLocation variable. This parameter has no direct relation to the location field in the system configuration element.
4. **snmp-enabled**—Set the SNMP system on this Oracle Enterprise Session Border Controller to **enabled** or **disabled**. By default, this parameter is set to enabled. The valid values are:
   - enabled | disabled
5. **enable-snmp-syslog-notify**—Set whether SNMP traps are sent when the system generates an alarm message. The SNMP agent sends a trap when an alarm is generated if the following conditions are met:
   - SNMP is enabled.
   - This field is enabled.
   - The syslog severity level is equal to or greater than the severity level configured in the SNMP Syslog Level field.

   The default is **disabled**. Valid values are:
   - **enabled** | **disabled**
6. **enable-snmp-monitor-traps**—When this parameter is enabled, the Oracle Enterprise Session Border Controller generates traps with unique trap-IDs for each syslog event. If this parameter is disabled , a single trap-ID is used for all events, with different values in the description string. The default is disabled. The valid values are:
   - enabled | disabled
7. **enable-snmp-auth-traps**—Set whether the SNMP authentication traps are enabled. If an SNMP request fails authentication because of an IPv4 address and SNMP community mismatch, the SNMP request will be rejected. This field determines if an SNMP trap will be sent in response to the authentication failure. The default is **disabled**. Valid values for this parameter are:
   - **enabled** | **disabled**
8. **enable-env-monitor-traps**—Set whether or not the SNMP environment monitor traps are enabled. Environment traps include main board PROM temperature, CPU voltage, power supplies, fan speeds, etc. The default is **disabled**. Valid values for this parameter are:
   - enabled | disabled
9. **snmp-syslog-his-table-length**—Set the length of the syslog trap history table. When a syslog message that meets the SNMP syslog level field criteria is generated and SNMP is enabled, the SNMP agent adds that message to a history table. This parameter indicates the number of entries the table can contain. The default is **1**. The valid range is:
   - Minimum—1
   - Maximum—500

   Once the last table entry is filled, the oldest entry will be overwritten with a new entry.
10. **snmp-syslog-level**—Set the log severity level threshold that will cause the syslog trap to be sent to an NMS. When this criteria is met and the appropriate SNMP trap is sent, an entry is written to the SNMP Syslog History Table. The default is **warning**. The following are valid values:
    - emergency | critical | major | minor | warning | notice | info | trace | debug | detail

SNMP Community Configuration

To configure SNMP communities:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type snmp-community and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# snmp-community
ACMEPACKET(snmp-community)#
```

From this point, you can set SNMP community parameters.

The following is an example what an SNMP Community configuration might look like. Parameters not described in this section are omitted below.

```
snmp-community
        community-name                  public
        access-mode                     READ-ONLY
        ip-addresses
                                        10.0.1.42
```

4. community-name—Set the SNMP community name of an active community where this Oracle Enterprise Session Border Controller can send or receive SNMP information. A community name value can also be used as a password to provide authentication, thereby limiting the NMSs that have access to this system. With this field, the SNMP agent provides trivial authentication based on the community name that is exchanged in plain text SNMP messages.
5. access-mode—Set the access level for all NMSs defined within this SNMP community. The access level determines the permissions that other NMS hosts can wield over this Oracle Enterprise Session Border Controller. The default is read-only. The valid values are:
   - read-only—allows GET requests.
   - read-write—allows both GET and SET requests.
6. ip-addresses—Set one or multiple IPv4 addresses that are valid within this SNMP community. These IPv4 addresses correspond with the IPv4 address of NMS applications that monitor or configure this Oracle Enterprise Session Border Controller. Include the IPv4 addresses of all servers where Element Management System is installed.

Trap Receiver Configuration

To configure trap receivers:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type trap-receiver and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# trap-receiver
ACMEPACKET(trap-receiver)#
```

From this point, you can set trap receivers.

The following is an example of a trap receiver configuration. Parameters not described in this section are omitted below.

```
trap-receiver
        ip-address                      10.0.1.42:162
        filter-level                    All
        community-name                  public
    user-list    jsmith,carolM,gcather
```

4. ip-address—Set the IPv4 address of an authorized NMS. This parameter is the IPv4 address of an NMS where traps are sent. If you do not specify a port number, the default SNMP trap port of 162 will be used.
5. filter-level—Set the filter level threshold that indicates the severity level at which a trap to be sent to this particular trap receiver. The default for this parameter is critical.

   Example: A trap with a severity level of Critical is generated, the SNMP agent will only send this trap to NMSs that are configured in a trap-receiver element and have a filter-level parameter of Critical.

   The following table maps Syslog and SNMP alarms to trap receiver filter levels.

Filter Level	Syslog Severity Level	(SNMP) Alarm Severity Level
Critical	Emergency (1) Critical (2)	Emergency Critical
Major	Emergency (1) Critical (2) Major (3)	Emergency Critical Major
Minor	Emergency (1) Critical (2) Major (3) Minor (4)	Emergency Critical Major Minor
All	Emergency (1) Critical (2) Major (3) Minor (4) Warning (5) Notice (6) Info (7) Trace (8) Debug (9)	Emergency Critical Major Minor Warning

   When configuring the trap-receiver element for use with the Element Management System (EMS), Oracle recommends that you set the filter-level parameter to All for that configuration element that includes EMS servers.
6. community-name—Set the community name to which this trap receiver belongs. This community must be defined in the SNMP community element.
7. user-list—For SNMPv3, specify a list of users that have authorized permissions to receive secure traps. Enter the user names as comma-separated values. For example:

```
ACMEPACKET(trap-receiver)# user-list jsmith,carolm,glather
```

Note: If instances of snmp-user-entry are configured, but no users are listed under user-list, a warning message is sent during a verify-config execution.

Media Supervision Traps

The Oracle Enterprise Session Border Controller, when functioning as a border gateway, will send the following trap when the media supervision timer has expired. This behavior is disabled by default, but can be enabled by changing the media-supervision-traps parameter to enabled in the media-manager configuration element.

```
apSysMgmtMediaSupervisionTimerExpTrap    NOTIFICATION-TYPE
      OBJECTS          { apSysMgmtCallId }
      STATUS            current
      DESCRIPTION
            " The trap will be generated when a media supervision timer
            has expired.  This behavior is disabled by default but may
            be enabled by changing the 'media-supervision-traps'
            parameter of the 'media-manager' configuration element.  The
            included object is the call identifer for the call which had
            the timer expire."
      ::= { apSystemManagementMonitors 34 }
```

The system does not send this trap when functioning as an integrated Oracle Enterprise Session Border Controller.

Syslog and Process Logs

Logging events is a critical part of diagnosing misconfigurations and optimizing operations. Oracle Enterprise Session Border Controllers can send both syslog and process log data to appropriate hosts for storage and analysis.

Overview

The Oracle Enterprise Session Border Controller generates two types of logs, syslogs and process logs. Syslogs conform to the standard used for logging servers and processes as defined in RFC 3164.

Process logs are Oracle proprietary logs. Process logs are generated on a per-task basis and are used mainly for debugging purposes. Because process logs are more data inclusive than syslogs, their contents usually encompass syslog log data.

Syslog and process log servers are both identified by an IPv4 address and port pair.

Process Log Messages

Process log messages are sent as UDP packets in the following format:

```
<file-name>:<log-message>
```

In this format, <filename> indicates the log filename and <log-message> indicates the full text of the log message as it would appear if it were written to the normal log file.

Syslog and Process Logs Configuration

This section describes how to configure syslog and process log servers.

To configure syslogs and process logs:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type system-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# system-config
   ACMEPACKET(system-config)#
   ```

   From this point, you can set process log parameters. Skip to the following process log configuration section.

4. Type syslog-server and press Enter. The system prompt changes to let you know that you can begin configuring individual syslog parameters

   ACMEPACKET(system-config)# syslog-server

ACMEPACKET(syslog-server)#

From this point, you can set syslog parameters. The following is an example what an syslog and process log configuration might look like. Parameters not described in this section are omitted below.

```
system-log-level                WARNING
syslog-server
        address                         172.15.44.12
        port                            514
        facility                        4
process-log-level               NOTICE
process-log-ip-address          0.0.0.0
process-log-port                0
```

Syslog Configuration

The Oracle Enterprise Session Border Controller supports multiple syslog servers. As the number of active syslog increases, the performance level of the Oracle Enterprise Session Border Controller may decrease. Therefore, we recommend configuring no more than 8 syslog servers.

Set the following parameters to configure syslog servers:

1. address—Set the IPv4 address of a syslog server.
2. port—Set the port portion of the syslog server. The default is 514.
3. facility—Set an integer to identify a user-defined facility value sent in every syslog message from the Oracle Enterprise Session Border Controller to the syslog server. This parameter is used only for identifying the source of this syslog message as coming from the Oracle Enterprise Session Border Controller. It is not identifying an OS daemon or process. The default value for this parameter is 4. RFC 3164 specifies valid facility values.

   In software release versions prior to Release 1.2, the Oracle Enterprise Session Border Controller would send all syslog messages with a facility marker of 4.
4. system-log-level—Set which log severity levels write to the system log (filename: acmelog). The default is WARNING. Valid values are:

   - EMERGENCY | CRITICAL | MAJOR | MINOR | WARNING | NOTICE | INFO | TRACE | DEBUG | DETAIL

Process Log Configuration

Set the following parameters to configure the process log server:

1. process-log-level—Set the starting log level all processes running on the system use. Each individual process running on the system has its own process log. The default is NOTICE. Valid values are:

   - EMERGENCY | CRITICAL | MAJOR | MINOR | WARNING | NOTICE | INFO | TRACE | DEBUG | DETAIL
2. process-log-ip-address—Set the IPv4 address of the process log server. The default 0.0.0.0, which causes log messages to be written to the normal log file.
3. process-log-port—Set the port number associated with the process log server. The default value for this parameter is 0, which causes log messages to be written to the normal log file. The valid range is:

   - Minimum—0
   - Maximum—65535.

Host Routes

Host routes let you insert entries into the Oracle Enterprise Session Border Controller's routing table. These routes affect traffic that originates at the Oracle Enterprise Session Border Controller’s host process. Host routes are used primarily for steering management traffic to the correct network.

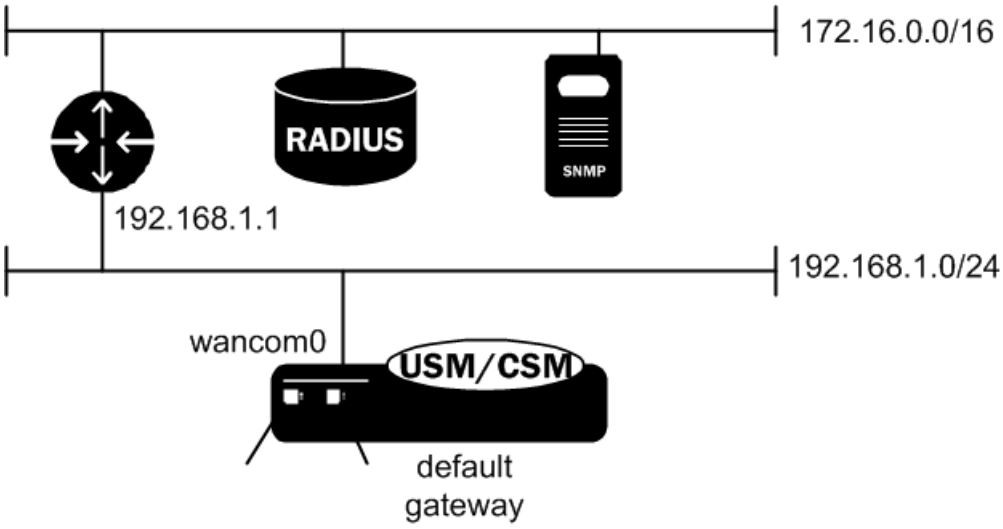
When traffic is destined for a network that is not explicitly defined on a Oracle Enterprise Session Border Controller, the default gateway (located in the system config) is used. If you try to route traffic to a specific destination that is not accessible through the default gateway, you need to add a host route. Host routes can be thought of as a default gateway override.

Certain SIP configurations require that the default gateway is located on a media interface. In this scenario, if management applications are located on a network connected to an administrative network, you will need to add a host route for management connectivity.

When source-based routing is used, the default gateway must exist on a media interface. Host routes might be needed to reach management applications connected to a management port in this kind of situation as well.

Host Routes Example

Because SIP signaling over media interfaces is enabled, the default gateway uses an IPv4 address assigned to a media interface. Maintenance services (SNMP and Radius) are located on a network connected to, but separate from, the 192.168.1.0/24 network on wancom0. In order to route Radius or SNMP traffic to an NMS (labeled as SNMP in the following example), a host route entry must be a part of the Oracle Enterprise Session Border Controller configuration. The host route tells the host how to reach the 172.16.0.0/16 network. The actual configuration is shown in the example in the next section of this guide.



Host Route Configuration

To configure a host route:

1. Access the **host-route** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# host-route
ACMEPACKET(host-route)#
```

2. **dest-network**—Set the IP address of the destination network that this host route points toward.
3. **netmask**—Set the netmask portion of the destination network for the route you are creating. The netmask is in dotted decimal notation.
4. **gateway**—Set the gateway that traffic destined for the address defined in the first two elements should use as its first hop.
5. Type **done** to save your configuration.

Setting Holidays in Local Policy

This section explains how to configure holidays on the Oracle Enterprise Session Border Controller.

You can define holidays that the Oracle Enterprise Session Border Controller recognizes. Holidays are used to identify a class of days on which a local policy is enacted. All configured holidays are referenced in the local-policy-attributes configuration subelement as an H in the days-of-week parameter. Because holidays are entered on a one-time basis per year, you must configure a new set of holidays yearly.

Holidays Configuration

To configure holidays:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type session-router-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-router-config
ACMEPACKET(session-router-config)#
```

4. Type holidays and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router-config)# holidays
ACMEPACKET(session-router-holidays)#
```

From this point, you can configure the holidays subelement. To view all holidays parameters, enter a ? at the system prompt.

```
holiday
                date                            2005-01-01
                description                     New Years Day
```

To configure a holiday, add an entry for the following parameters in the holidays element:

5. date—Enter the holiday's date in YYYY-MM-DD format.
6. description—Enter a short description for the holiday you are configuring. If the description contains words separated by spaces, enter the full description surrounded by quotation marks.

Enhanced Control of UDP and TCP Ports

This section explains how to configure the Oracle Enterprise Session Border Controller for finer control of the set of UDP and TCP ports that on which the Oracle Enterprise Session Border Controller provides services. The settings you can configure have an impact on:

- UDP/TCP port 111 (the RPC services port), which is disabled on Oracle Enterprise Session Border Controller startup but can be enabled in the boot parameters
- TCP ports 3000 (used when notify commands are issued remotely, i.e. via an element management system) and 3001 (used for remote configuration, i.e. via an element management system), which can now be enabled or disabled in the system configuration

Neither configuration for these features is covered by RTC, so you must reboot your Oracle Enterprise Session Border Controller for changes to take effect. Be aware that rebooting can cause system downtime, and plan accordingly.

Port 111 Configuration

To enable port 111 using Oracle Enterprise Session Border Controller boot parameters:

1. In Superuser mode, type configure terminal and press Enter

```
ACMEPACKET# configure terminal
```

2. To enter the boot parameters so that you can configure them, type bootparam and press Enter.

```
ACMEPACKET(configure)# bootparam
```

3. Press Enter to scroll through the list of boot parameters until you reach the setting for flags.

   To set this value correctly, you need to add the value 0x200000 to your existing flag setting in the boot parameters. In the example below, the existing flag value is 0x30008. When the value 0x200000 is added, the result is 0x230008. The result is the value that you need to set.

   When you reach the flag setting, type the value representing the flags you need (0x230008 in the example below) and press Enter. Continue to press Enter to finish scrolling through the rest of the boot parameters.

```
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device              : wancom0
processor number     : 0
host name     : acmepacket8
file name     : /tffs0/sd220p9.gz
inet on ethernet (e)    : 10.0.1.57:ffff0000
inet on backplane (b)    : 0.0.0.0
host inet (h)            : 10.0.1.5
gateway inet (g)         : 10.0.0.1
user (u)                 : user
ftp password (pw)        : password
flags (f)                : 0x30008 0x230008
target name (tn)         : acmesystem
startup script (s)       : 0
other (o)                :
NOTE: These changed parameters will not go into effect until reboot. Also,
be aware that some boot parameters may also be changed through the PHY and
Network Interface Configurations.
ACMEPACKET(configure)#
```

4. Type exit to return to the main Superuser menu so that you can reboot your Oracle Enterprise Session Border Controller and apply the settings you have entered.

```
ACMEPACKET(configure)# exit
```

5. Reboot your Oracle Enterprise Session Border Controller. Type a y and press Enter to reboot.

```
ACMEPACKET# reboot
----------------------------------------
WARNING: you are about to reboot this SD!
----------------------------------------
Reboot this SD [y/n]?:y
```

Port 3000 and 3001 Configuration

To control TCP ports 3000 and 3001 in the system configuration:

1. Access the **security-config** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# security-config
ACMEPACKET(security-config)#
```

2. Type **select** to begin editing the **system-config** object.

```
ACMEPACKET(system-config)# select
ACMEPACKET(system-config)#
```

3. The parameter controlling ports 3000 and 3001 is called remote-control, and its default is enabled. To disable the ports, set this parameter to disabled.

```
ACMEPACKET(system-config)# remote-control disabled
```

4. Type **done** to save your configuration.
5. Reboot your Oracle Enterprise Session Border Controller. Type a y and press Enter to reboot.

```
ACMEPACKET# reboot
----------------------------------------
WARNING: you are about to reboot this SD!
----------------------------------------
Reboot this SD [y/n]?:y
```

DNS Transaction Timeout

This section explains how to configure the DNS transaction timeout interval on a per network-interface basis. You can currently configure the Oracle Enterprise Session Border Controller with a primary and two optional backup DNS servers. The Oracle Enterprise Session Border Controller queries the primary DNS server and upon not receiving a response within the configured number of seconds, queries the backup1 DNS server and if that times out as well, then contacts the backup2 DNS server.

Retransmission Logic

The retransmission of DNS queries is controlled by three timers. These timers are derived from the configured DNS timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the Oracle Enterprise Session Border Controller should retransmit 3 times before timing out to give the server a chance to respond.

- Init-timer is the initial retransmission interval. If a response to a query is not received within this interval, the query is retransmitted. To safeguard from performance degradation, the minimum value allowed for this timer is 250 milliseconds.
- Max-timer is the maximum retransmission interval. The interval is doubled after every retransmission. If the resulting retransmission interval is greater than the value of max-timer, it is set to the max-timer value.
- Expire-timer: is the query expiration timer. If a response is not received for a query and its retransmissions within this interval, the server will be considered non-responsive and the next server in the list will be tried.

The following examples show different timeout values and the corresponding timers derived from them.

```
timeout >= 3 seconds
Init-timer = Timeout/11
Max-Timer = 4 * Init-timer
Expire-Timer = Timeout
timeout = 1 second
Init-Timer = 250 ms
Max-Timer = 250 ms
Expire-Timer = 1 sec
timeout = 2 seconds
Init-Timer = 250 ms
Max-Timer = 650 ms
Expire-Timer = 2sec
```

DNS Transaction Timeout Configuration

To configure DNS transaction timeout:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type network-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)#
```

From this point, you can configure network interface parameters. To view all network interface parameters, enter a ? at the system prompt.

4. dns-timeout—Enter the total time in seconds you want to elapse before a query (and its retransmissions) sent to a DNS server would timeout. The default is 11 seconds. The valid range is:

   - Minimum—1
   - Maximum—999999999.

   If a query sent to the primary DNS server times out, the backup1 DNS server is queried. If the query times out after the same period of time elapses, the query continues on to the backup2 DNS server.

5. Save and activate your configuration.

Persistent Protocol Tracing

This section explains how to configure persistent protocol tracing to capture specific SIP and MGCP protocol message logs and persistently send them off the Oracle Enterprise Session Border Controller, even after rebooting the system. This feature is not applicable to log for H.323 or IWF.

About Persistent Protocol Tracing

You can configure sending protocol message logs off of the Oracle Enterprise Session Border Controller, and have that persist after a reboot. You no longer have to manually issue the notify command each time you reboot.

To support persistent protocol tracing, you configure the following system-config parameters:

- call-trace—Enable/disable protocol message tracing (currently only sipmsg.log and alg.log) regardless of the process-log-level setting. If the process-log-level is set to trace or debug, call-trace will not disable.
- internal-trace—Enable/disable internal ACP message tracing for all processes, regardless of process-log-level setting. This applies to all *.log (internal ACP message exchange) files other than sipmsg.log and alg.log. If the process-log-level is set to trace or debug, call-trace will not disable.
- log-filter—Determine what combination of protocol traces and logs are sent to the log server defined by the process-log-ip parameter value. You can also fork the traces and logs, meaning that you keep trace and log information in local storage as well as sending it to the server. You can set this parameter to any of the following values: none, traces, traces-fork, logs, logs, all, or all-fork.

  The Oracle Enterprise Session Border Controller uses the value of this parameter in conjunction with the process-log-ip and process-log-port values to determine what information to send. If you have configured the proc-log-ip and proc-log-port parameters, choosing traces sends just the trace information (provided they are turned on), logs sends only process logs (log.*), and all sends everything (which is the default).

About the Logs

When you configure persistent protocol tracing, you affect the following types of logs.

Note: Enabling logs can have an impact on Oracle Enterprise Session Border Controller performance.

Process Logs

Events are logged to a process log flow from tasks and are specific to a single process running on the Oracle Enterprise Session Border Controller. By default they are placed into individual files associated with each process with the following name format:

log.<taskname>

By setting the new log-filter parameter, you can have the logs sent to a remote log server (if configured). If you set log-filter to logs or all, the logs are sent to the log server. Otherwise, the logs are still captured at the level the process-

log-level parameter is set to, but the results are stored on the Oracle Enterprise Session Border Controller's local storage.

Communication Logs

These are the communication logs between processes and system management. The logs are usually named <name>.log, with <name> being the process name. For example, sipd.log.

This class of log is configured by the new internal-trace parameter.

Protocol Trace Logs

The only protocol trace logs included at this time are sipmsg.log for SIP and alg.log for MGCP. (The H.323 system tracing is not currently included.) All of the logs enabled with the call–trace parameter are sent to remote log servers, if you also set the log-filter parameter to logs or all.

Persistent Protocol Tracing Configuration

Before you configure persistent protocol tracing, ensure you have configured the process logs by setting the system configuration's process-log-ip parameter.

To configure persistent protocol tracing:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type system-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

4. call-trace—Set to enabled to enable protocol message tracing for sipmsg.log for SIP and alg.log for MGCP. The default is disabled. The valid values are:
   - enabled | disabled
5. internal-trace—Set to enabled to enable internal ACP message tracing for all processes. The default is disabled. The valid values are:
   - enabled | disabled
6. log-filter—Choose the appropriate setting for how you want to send and/or store trace information and process logs. The valid values are:
   - none—No information will be sent or stored.
   - traces—Sends the trace information to both the log server; includes <name>.log files that contain information about the Oracle Enterprise Session Border Controller's internal communication processes (<name> is the name of the internal process)
   - traces-fork—Sends the trace information to both the log server and also keeps it in local storage; includes <name>.log files that contain information about the Oracle Enterprise Session Border Controller's internal communication processes (<name> is the name of the internal process)
   - logs—Sends the process logs to both the log server; includes log.* files, which are Oracle Enterprise Session Border Controller process logs
   - logs-fork—Sends the process logs to both the log server and also keeps it in local storage; includes log.* files, which are Oracle Enterprise Session Border Controller process logs
   - all—Sends all logs to the log servers that you configure
   - all-fork—Sends all logs to the log servers that you configure, and it also keeps the logs in local storage
7. Save and activate your configuration.

System Access Control

You can configure a system access control list (ACL) for your Oracle Enterprise Session Border Controller that determines what traffic the Oracle Enterprise Session Border Controller allows over its management interface (wancom0). By specifying who has access to the Oracle Enterprise Session Border Controller via the management interface, you can provide DoS protection for this interface.

Using a list of IP addresses and subnets that are allowable as packet sources, you can configure what traffic the Oracle Enterprise Session Border Controller accepts and what it denies. All IP packets arriving on the management interface are subject; if it does not match your configuration for system ACL, then the Oracle Enterprise Session Border Controller drops it.

Note: All IP addresses configured in the SNMP community table are automatically permitted.

Adding an ACL for the Management Interface

The new subconfiguration system-access-list is now part of the system configuration, and its model is similar to host routes. For each entry, you must define an IP destination address and mask; you can specify either the individual host or a unique subnet.

If you do not configure this list, then there will be no ACL/DoS protection for the Oracle Enterprise Session Border Controller's management interface.

You access the system-access-list via system path, where you set an IP address and netmask. You can configure multiple system ACLs using this configuration.

To add an ACL for the management interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system-access-list and press Enter.

```
ACMEPACKET(system)# system-access-list
ACMEPACKET(system-access-list)#
```

4. source-address—Enter the IP address representing for the source network for which you want to allow traffic over the management interface.
5. netmask—Enter the netmask portion of the source network for the traffic you want to allow. The netmask is in dotted decimal notation.

Notes on Deleting System ACLs

If you delete a system ACL from your configuration, the Oracle Enterprise Session Border Controller checks whether or not there are any active FTP or Telnet client was granted access when the entry was being removed. If such a client were active during ACL removal, the Oracle Enterprise Session Border Controller would warn you about the condition and ask you to confirm the deletion. If you confirm the deletion, then the Oracle Enterprise Session Border Controller's session with the active client is suspended.

The following example shows you how the warning message and confirmation appear. For this example, and ACLI has been deleted, and the user is activating the configuration that reflects the change.

```
ACMEPACKET # activate-config
Object deleted will cause service disruption:
 system-access-list: identifier=172.30.0.24
 ** WARNING: Removal of this system-ACL entry will result
             in the lockout of a current FTP client
```

```
Changes could affect service, continue (y/n) y
Activate-Config received, processing.
```

System TCP Keepalive Settings

You can configure the Oracle Enterprise Session Border Controller to control TCP connections by setting:

- The amount of time the TCP connection is idle before the Oracle Enterprise Session Border Controller starts sending keepalive messages to the remote peer
- The number of keepalive packets the Oracle Enterprise Session Border Controller sends before terminating the TCP connection

If TCP keepalive fails, then the Oracle Enterprise Session Border Controller will drop the call associated with that TCP connection.

In the ALCI, a configured set of network parameters appears as follows:

```
network-parameters
        tcp-keepinit-timer              75
        tcp-keepalive-count             4
        tcp-keepalive-idle-timer        400
        tcp-keepalive-interval-timer    75
        tcp-keepalive-mode              0
```

Then you apply these on a per-interface basis. For example, the H.323 interface (stack) configuration allows you to enable or disabled use of the network parameters settings.

System TCP Keepalive Configuration

TCP setting are global, and then enabled or disabled on a per-interface basis.

To configure TCP keepalive parameters on your Oracle Enterprise Session Border Controller:

Note: If you want to use the default values for TCP keepalive, you do not need to take Steps 1 through 4. You can simply set the TCP keepalive function in the H.323 stack configuration, and the defaults for network parameters will be applied.

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-related configurations.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type network-parameters and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# network-parameters
   ACMEPACKET(network-parameters)#
   ```

4. tcp-keepinit-timer—If a TCP connection cannot be established within some amount of time, TCP will time out the connect attempt. It can be used to set the initial timeout period for a given socket, and specifies the number of seconds to wait before the connect attempt is timed out. For passive connections, this value is inherited from the listening socket. The default is 75. The valid range is:

   - Minimum—0
   - Maximum—999999999.

5. tcp-keepalive-count—Enter the number of packets the Oracle Enterprise Session Border Controller sends to the remote peer before it terminates the TCP connection. The default is 8. The valid range is:

   - Minimum—0
   - Maximum—223-1

6. tcp-keepalive-idle-timer—Enter the number of seconds of idle time before TCP keepalive messages are sent to the remote peer if the SO-KEEPALIVE option is set. This option is set via the h323-stack configuration element. The default is 7200. The valid range is:
   - Minimum—30
   - Maximum—7200
7. tcp-keepalive-interval-timer—When the SO_KEEPALIVE option is enabled, TCP probes a connection that has been idle for some amount of time. If the remote system does not respond to a keepalive probe, TCP retransmits the probe after a set amount of time. This parameter specifies the number of seconds to wait before retransmitting a keepalive probe. The default value is 75 seconds. The valid range is:
   - Minimum—15
   - Maximum—75
8. tcp-keepalive-mode—Set the TCP keepalive response sequence number. The default is 0. The valid values are:
   - 0—The sequence number is sent un-incremented
   - 1—The number is incremented
   - 2—No packets are sent

Configurable TCP Timers

You can configure your Oracle Enterprise Session Border Controller to detect failed TCP connections more quickly so that data can be transmitted via an alternate connection before timers expire. Across all protocols, you can now control the following for TCP:

- Connection establishment
- Data retransmission
- Timer for idle connections

These capabilities all involve configuring an options parameter that appears in the network parameters configuration.

Configuring TCP Connection Establishment

To establish connections, TCP uses a three-way handshake during which two peers exchange TCP SYN messages to request and confirm the active open connection. In attempting this connection, one peer retransmits the SYN messages for a defined period of time if it does not receive acknowledgement from the terminating peer. You can configure the amount of time in seconds between the retries as well as how long (in seconds) the peer will keep retransmitting the messages.

You set two new options in the network parameters configuration to specify these amounts of time: atcp-syn-rxmt-interval and atcp-syn-rxmt-maxtime.

Note that for all configured options, any values entered outside of the valid range are silently ignored during configuration and generate a log when you enter the activate command.

To configure TCP connection establishment:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type network-parameters and press Enter.

```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

4. options—Set the options parameter by typing options, a Space, the option name atcp-syn-rxmt-interval=x (where x is a value in seconds between 2 and 10) with a plus sign in front of it. Then press Enter. This value will be used

as the interval between TCP SYN messages when the Oracle Enterprise Session Border Controller is trying to establish a connection with a remote peer.

Now enter a second option to set the maximum time for trying to establish a TCP connection. Set the options parameter by typing options, a Space, the option name atcp-syn-rxmt-maxtime=x (where x is a value in seconds between 5 and 75) with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(network-parameters)# options +atcp-syn-rxmt-interval=5
ACMEPACKET(network-parameters)# options +atcp-syn-rxmt-maxtime=30
```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

Note: atcp-syn-rxmt-maxtime=x option is equivalent to the tcp-keepinit-timer parameter, but only affects ATCP.

5. Save and activate your configuration.

Configuring TCP Data Retransmission

TCP is considered reliable in part because it requires that entities receiving data must acknowledge transmitted segments. If data segments go unacknowledged, then they are retransmitted until they are finally acknowledged or until the maximum number of retries has been reached. You can control both the number of times the Oracle Enterprise Session Border Controller tries to retransmit unacknowledged segments and the periodic interval (how often) at which retransmissions occur.

You set two new options in the network parameters configuration to specify how many retransmissions are allowed and for how long: atcp-rxmt-interval and atcp-rxmt-count.

To configure TCP data retransmission:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type network-parameters and press Enter.

```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

4. options—Set the options parameter by typing options, a Space, the option name atcp-rxmt-interval=x (where x is a value in seconds between 2 and 60) with a plus sign in front of it. Then press Enter. This value will be used as the interval between retransmission of TCP data segments that have not been acknowledged.

   Now enter a second option to set the number of times the Oracle Enterprise Session Border Controller will retransmit a data segment before it declares the connection failed. Set the options parameter by typing options, a Space, the option name atcp-rxmt-count=x (where x is a value between 4 and 12 representing how many retransmissions you want to enable) with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(network-parameters)# options +atcp-rxmt-interval=30
ACMEPACKET(network-parameters)# options +atcp-rxmt-count=6
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Timer for Idle Connections

When enabled to do so, the Oracle Enterprise Session Border Controller monitors inbound TCP connections for inactivity. These are inbound connections that the remote peer initiated, meaning that the remote peer sent the first

SYN message. You can configure a timer that sets the maximum amount of idle time for a connection before the Oracle Enterprise Session Border Controller consider the connection inactive. Once the timer expires and the connection is deemed inactive, the Oracle Enterprise Session Border Controller sends a TCP RST message to the remote peer.

To configure the timer for TCP idle connections:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type network-parameters and press Enter.

```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

4. options—Set the options parameter by typing options, a Space, the option name atcp-idle-timer=x (where x is a value in seconds between 120 and 7200) with a plus sign in front of it. Then press Enter. This value will be used to measure the activity of TCP connections; when the inactivity on a TCP connection reaches this value in seconds, the Oracle Enterprise Session Border Controllerdeclares it inactive and drops the session.

```
ACMEPACKET(network-parameters)# options +atcp-idle-timer=900
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Historical Data Recording (HDR)

Updated HDR and HDR configuration information resides in the Net-Net® C-Series Historical Data Recording (HDR) Resource Guide Version C6.3.0, 400-0141-63 (Net-Net 4000 S-CX6.3.0 HDR Resource Guide.pdf). This document is available with the complete Version S-CX6.3.0 documentation set.

Packet Trace

Oracle Enterprise Session Border Controller Release 5.0 introduces the packet trace feature to the Oracle Enterprise Session Border Controller's capabilities. When you enable this feature, the Oracle Enterprise Session Border Controller can mirror any communication between two endpoints, or between itself and a specific endpoint. To accomplish this, the Oracle Enterprise Session Border Controller replicates the packets sent and received, and can then send them to a trace server that you designate. Using the trace server, you can display the packets on software protocol analyzer. Currently, the Oracle Enterprise Session Border Controller supports:

- One configurable trace server (on which you have installed your software protocol analyzer)
- Sixteen concurrent endpoint traces

To use this feature, you configure a trace server on the Oracle Enterprise Session Border Controller so that it knows where to send the mirrored packets. Once the trace server is configured, the Oracle Enterprise Session Border Controller uses one of its internally configured IP addresses (such as one for a SIP interface or for an H.323 interface) on which to base the trace.

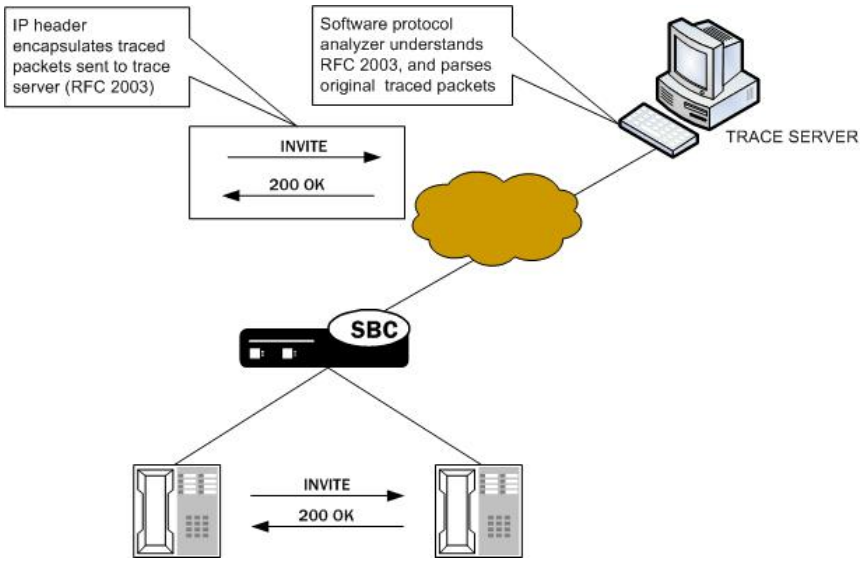
You start a packet trace using the ACLI Superuser command packet-trace start, enter with these pieces of information:

- Network interface—The name of the network interface on the Oracle Enterprise Session Border Controller from which you want to trace packets; this value can be entered either as a name alone or as a name and subport identifier value (name:subportid)

- IP address—IP address of the endpoint to and from which the Oracle Enterprise Session Border Controller will mirror calls
- Local port number—Optional parameter; Layer 4 port number on which the Oracle Enterprise Session Border Controller receives and from which it sends; if no port is specified or if it is set to 0, then all ports will be traced
- Remote port number—Optional parameter; Layer 4 port number to which the Oracle Enterprise Session Border Controller sends and from which it receives; if no port is specified or if it is set to 0, then all ports will be traced

Once the trace is initiated, the Oracle Enterprise Session Border Controller duplicates all packets sent to and from the endpoint identified by the IP address that are sent or received on the specified Oracle Enterprise Session Border Controller network interface.

The Oracle Enterprise Session Border Controller then encapsulates the original packets in accordance with RFC 2003 (IP Encapsulation within IP); it adds the requisite headers, and the payload contains the original packet trace with the Layer 2 header removed. Since software protocol analyzers understand RFC 2003, they can easily parse the original traced packets. In order to see only packet traces information in your software protocol analyzer, you can use a capture filter; for example, the Ethereal/Wireshark syntax is ip proto 4.



It is possible that—for large frames—when the Oracle Enterprise Session Border Controller performs the steps to comply with RFC 2003 by adding the requisite header, the resulting packet might exceed Ethernet maximum transmission unit (MTU). This could result in packets being dropped by external network devices, but widespread support for jumbo frames should mitigate this possibility.

If the Oracle Enterprise Session Border Controller either receives or transmits IP fragments during a packet trace, then it will only trace the first fragment. The first fragment is likely to be a maximum-sized Ethernet frame.

The Oracle Enterprise Session Border Controller continues to conduct the packet trace and send the replicated information to the trace server until you instruct it to stop. You stop a packet trace with the ACLI packet-trace stop command. With this command, you can stop either an individual packet trace or all packet traces that the Oracle Enterprise Session Border Controller is currently conducting.

Packet Trace Scenarios

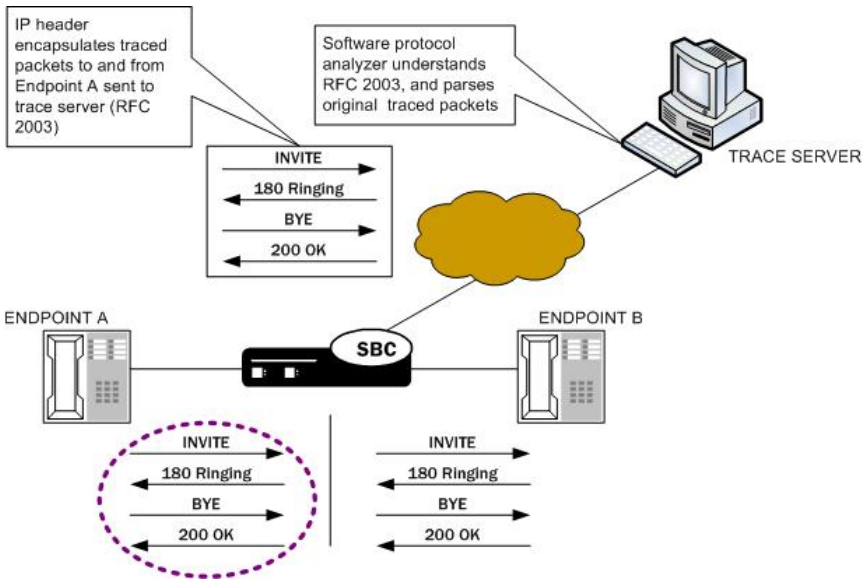
This section describes three possible ways that you might use the packet trace feature. You can examine communications sent to and from one endpoint, sent between two endpoints, or sent between ingress and egress Oracle Enterprise Session Border Controller interfaces to endpoints.

Packet Trace for One Endpoint

When you use the packet-trace-state command, the Oracle Enterprise Session Border Controller sets up packet tracing for one endpoint. The Oracle Enterprise Session Border Controller collects and replicates the packets to and from one endpoint. To enable this kind of trace, you set up one packet trace using the packet-trace start command.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Endpoint A>
```



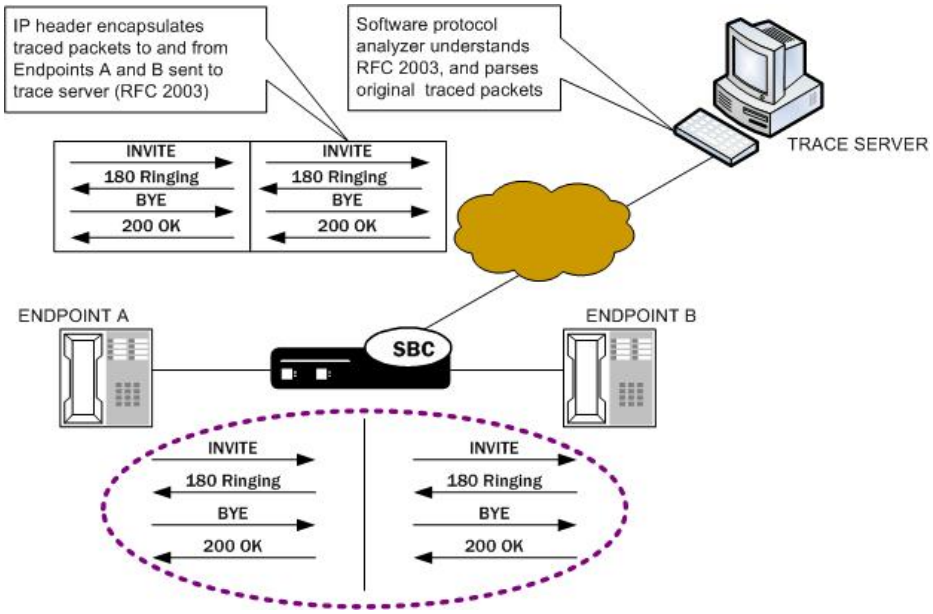
Packet Trace for Both Call Legs

If you want to trace both sides (both call legs), then you must set up individual traces for each endpoint—meaning that you would initiate two packet traces. The results of the trace will give you the communications both call legs for the communication exchanged between the endpoints you specify.

If you initiate a packet trace for both endpoints that captures both signaling and media, the signaling will be captured as usual. However, RTP will only be traced for the ingress call leg. This is because the Oracle Enterprise Session Border Controller performs NAT on the RTP, which means it cannot be captured on the egress call leg.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Endpoint A>
ACMEPACKET# packet-trace start F02 <IP address of Endpoint B>
```

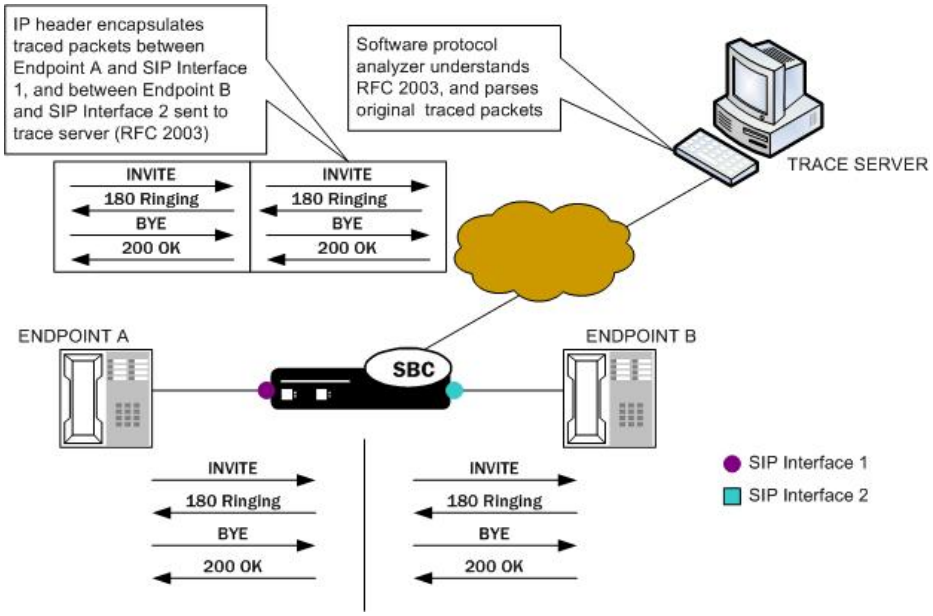


Packet Trace for a Oracle Enterprise Session Border Controller Signaling Address

You can perform a packet trace for addresses internal to the Oracle Enterprise Session Border Controller; this can be the address, for example, of a SIP or an H.323 interface. Using signaling interface addresses puts the emphasis on the Oracle Enterprise Session Border Controller rather than on the endpoints by allowing you to view traffic from specified interfaces.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Net-Net SBC interface1>
ACMEPACKET# packet-trace start F02 <IP address of Net-Net SBC interface2>
```



packet-trace

The packet-trace command starts or stops packet tracing on the Oracle Enterprise Session Border Controller. Once the trace is initiated, the Oracle Enterprise Session Border Controller duplicates all packets sent to and from the endpoint identified by the IP address that are sent or received on the specified Oracle Enterprise Session Border Controller network interface.

Syntax

```
packet-trace <start> <stop>
```

Arguments

<start> - Start **packet-tracing** on the Oracle Enterprise Session Border Controller. Once the trace is initiated, the Oracle Enterprise Session Border Controller duplicates all packets sent to and from the endpoint identified by the IP address that are sent or received on the specified Oracle Enterprise Session Border Controller network interface.

- network-interface—The name of the network interface on the Oracle Enterprise Session Border Controller from which you want to trace packets; this value can be entered as either a name alone or as a name and subport identifier value (name:subportid)
- ip-address—IP address of the endpoint to and from which the Oracle Enterprise Session Border Controller will mirror calls
- local-port—Layer 4 port number on which the Oracle Enterprise Session Border Controller receives and from which it sends. This is an optional parameter; if no port is specified or if it is set to 0, then all ports will be traced.
- remote-port—Layer 4 port to which the Oracle Enterprise Session Border Controller sends and from which it receives. This is an optional parameter; if no port is specified or if it is set to 0, then all ports are traced.

<stop> - Manually stop packet tracing on the Oracle Enterprise Session Border Controller. With this command you can either stop an individual packet trace or all packet traces that the Oracle Enterprise Session Border Controller is currently conducting.

- network-interface—The name of the network interface on the Oracle Enterprise Session Border Controller from which you want to stop packet tracing. This value can be entered either as a name alone or as a name and subport identifier value (name:subportid).
- ip-address—IP address of the endpoint to and from which you want the Oracle Enterprise Session Border Controller to stop mirroring calls.
- local-port—Layer 4 port number on which to stop from receiving and sending. This is an optional parameter; if no port is specified or if it is set to 0, then all port tracing will be stopped.
- remote-port—Layer 4 port number on which to stop the Oracle Enterprise Session Border Controller from receiving and sending. This is an optional parameter; if no port is specified or if it is set to 0, then all port tracing will be stopped.

Mode

Superuser

Release

First appearance: 5.0

Example

```
ACMEPACKET# packet-trace start public:0 111.0.12.5
```

Packet Trace Configuration

There are three steps you can take when you use the packet trace feature:

- Configuring the Oracle Enterprise Session Border Controller with the trace server information so that the Oracle Enterprise Session Border Controller knows where to send replicated data
- Setting up the capture filter ip proto 4 in your software protocol analyzer if you only want to see the results of the Oracle Enterprise Session Border Controller packet trace(s)
- Starting a packet trace
- Stopping a packet trace

This section provides information about how to perform all three tasks.

Configuring a Trace Server

You need to configure a trace server on the Oracle Enterprise Session Border Controller; this is the device to which the Oracle Enterprise Session Border Controller sends replicated data. The Oracle Enterprise Session Border Controller supports one trace server.

To configure a trace server on your Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter.

   ```
   ACMEPACKET(configure)# system
   ACMEPACKET(system)#
   ```

3. Enter capture-receiver and press Enter.

   ```
   ACMEPACKET(system)# capture-receiver
   ACMEPACKET(capture receiver)#
   ```

4. state—Type enabled so that you can use the trace server to which you want to send the mirrored packets for calls you are packet tracing. The default is disabled. The valid values are:

- enabled | disabled

  Disable capture receivers you are not actively using for traces to prevent potential service outages caused by the capture's system resource utilization.

5. address—Enter the IP address of the trace server; there is no default.
6. network-interface—Enter the name and subport of the Oracle Enterprise Session Border Controller network interface from which the Oracle Enterprise Session Border Controller is to send mirrored packets. Your entry needs to take the form name:subport. The default is :0.
7. Save and activate your configuration.

Starting a Packet Trace

You use the start a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- IP address to be traced; if you do not enter local and/or remote ports when you start the trace, the Oracle Enterprise Session Border Controller will trace all ports
- (Optional) Local UDP/TCP port on which the Oracle Enterprise Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced; you cannot enter the remote port without specifying a local port

To start a packet trace with local and remote ports specified:

Enter the ACLI packet-trace command followed by a Space, and the word start. After another Space, type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

```
ACMEPACKET# packet-trace start core:0 192.168.10.99 5060 5060
Trace started for 192.168.10.99
```

Stopping a Packet Trace

You use the stop a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- IP address to be traced
- (Optional) Local UDP/TCP port on which the Oracle Enterprise Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced

If the packet trace you want to stop has no entries for local and/or remote ports, then you do not have to specify them.

You have two options when stopping a packet trace:

1. To stop a packet trace with local and remote ports specified, enter the ACLI packet-trace command followed by a Space, and the word stop. After another Space, type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

   ```
   ACMEPACKET# packet-trace stop core:0 192.168.10.99 5060 5060
   ```

2. To stop all packet traces on the Oracle Enterprise Session Border Controller, enter the ACLI packet-trace command followed by a Space, and the word stop. After another Space, type the word all and press Enter.

   ```
   ACMEPACKET# packet-trace stop all
   ```

RAMdrive Log Cleaner

The RAMdrive log cleaner allows the Oracle Enterprise Session Border Controller to remove log files proactively and thereby avoid situations where running low on RAMdrive space is a danger. Because even a small amount of logging can consume a considerable space, you might want to enable the RAMdrive log cleaner.

The RAMdrive cleaner periodically checks the remaining free space in the RAMdrive and, depending on the configured threshold, performs a full check on the /ramdrv/logs directory. During the full check, the RAMdrive cleaner determines the total space logs files are using and deletes log files that exceed the configured maximum lifetime. In addition, if the cleaner finds that the maximum log space has been exceeded or the minimum free space is not sufficient, it deletes older log files until the thresholds are met.

Not all log files, however, are as active as others. This condition affects which log files the log cleaner deletes to create more space in RAMdrive. More active log files rotate through the system more rapidly. So, if the log cleaner were to delete the oldest of these active files, it might not delete less active logs files that could be older than the active ones. The log cleaner thus deletes files that are truly older, be they active or inactive.

Applicable Settings

In the system configuration, you establish a group of settings in the options parameter that control the log cleaner's behavior:

- ramdrv-log-min-free—Minimum percent of free space required when rotating log files.

  When the amount of free space on the RAMdrive falls below this value, the log cleaner deletes the oldest copy of the log file. The log cleaner also uses this setting when performing period cleaning.
- ramdrv-log-max-usage—Maximum percent of the RAMdrive the log files can use.

  The log cleaner removes old log files to maintain this threshold.
- ramdrv-log-min-check—Minimum percent of free space on the RAMdrive that triggers the log cleaner to perform a full check of log files.
- ramdrv-min-log-check—Minimum time (in seconds) between log cleaner checks.
- ramdrv-max-log-check—Maximum time (in seconds) between log cleaner checks. This value must be greater than or equal to the ramdrv-min-log-check.
- ramdrv-log-lifetime—Maximum lifetime (in days) for log files. You give logs unlimited lifetime by entering a value of 0.

Clean-Up Procedure

The log cleaner checks the amount of space remaining in the RAMdrive and performs a full check of the logs directory when:

- Free space is less than the minimum percent of the RAMdrive that triggers a full check of log files
- The amount of free space has changed by more than 5% of the RAMdrive capacity since the last full check
- A full check of the logs directory has not been performed in the last hour

When it checks the logs directory, the log cleaner inventories the collected log files. It identifies each files as one of these types:

- Process log—Files beginning with log.
- Internal trace file—A <task>.log file
- Protocol trace file—Call trace including sipmsg.log, dns.log, sipddns.log, and alg.log
- CDR file—File beginning with cdr

Next, the log cleaner determines the age of the log files using the number of seconds since the log files were created. Then it orders the files from oldest to newest. The age adjusts such that it always increases as the log file sequence number (a suffix added by file rotation) increases. The log cleaner applies an additional weighting factor to produce a weighted age that favors the preservation of protocol traces files over internal trace files, and internal trace files over

process log files. The base log file and CDR files are excluded from the age list and so will not be deleted; the accounting configuration controls CDR file aging.

With the age list constructed, the log cleaner examines the list from highest weighted age to lowest. If the actual file age exceeds the RAMdrive maximum log lifetime, the log cleaner deletes it. Otherwise, the log cleaner deletes files until the maximum percent of RAMdrive that logs can use is no longer exceeded and until the minimum percent of free space required when rotating logs is available.

Clean-Up Frequency

The minimum free space that triggers a full check of log files and the maximum time between log file checks control how often the log cleaner performs the clean-up procedure. When it completes the procedure, the log cleaner determines the time interval until the next required clean-up based on the RAMdrive's state.

If a clean-up results in the deletion of one or more log files or if certain thresholds are exceeded, frequency is based on the minimum time between log cleaner checks. Otherwise, the system gradually increases the interval up to the maximum time between log cleaner checks. The system increases the interval by one-quarter of the difference between the minimum and maximum interval, but not greater than one-half the minimum interval or smaller than 10 seconds. For example, using the default values, the interval would be increased by 30 seconds.

RAMdrive Log Cleaner Configuration

You configure the log cleaner's operating parameters and thresholds in the system configuration. Note that none of these settings is RTC-supported, so you must reboot your Oracle Enterprise Session Border Controller in order for them to take effect. If you are using this feature on an HA node, however, you can add this feature without impact to service by activating the configuration, rebooting the standby, switching over to make the newly booted standby active, and then rebooting the newly standby system.

Unlike other values for options parameters, the Oracle Enterprise Session Border Controller validates these setting when entered using the ACLI. If any single value is invalid, they all revert to their default values.

To configure the RAMdrive log cleaner:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system-config and press Enter.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

4. options—Set the options parameter by typing options, a Space, <option name>=X (where X is the value you want to use) with a plus sign in front of it. Then press Enter.

   Remember that if any of your settings are invalid, the Oracle Enterprise Session Border Controller changes the entire group of these options back to their default settings.

Option Name	Description
ramdrv-log-min-free	Minimum percent of free space required when rotating log files. When the amount of free space on the RAMdrive falls below this value, the log cleaner deletes the oldest copy of the log file. The log cleaner also uses this setting when performing period cleaning. Default=40; Minimum=15; Maximum=75
ramdrv-log-max-usage	Maximum percent of the RAMdrive the log files can use.

Option Name	Description
	The log cleaner removes old log files to maintain this threshold. Default=40; Minimum=15; Maximum=75
ramdrv-log-min-check	Minimum percent of free space on the RAMdrive that triggers the log cleaner to perform a full check of log files. Default=50; Minimum=25; Maximum=75
ramdrv-min-log-check	Maximum time (in seconds) between log cleaner checks. This value must be greater than or equal to the ramdrv-min-log-check. Default=180; Minimum=40; Maximum=1800
ramdrv--log-lifetime	Maximum lifetime (in days) for log files. You give logs unlimited lifetime by entering a value of 0. Default=30; Minimum=2; Maximum=9999

```
ACMEPACKET(system-config)# options +ramdrv-log-min-free=50
ACMEPACKET(system-config)# options +ramdrv-log-max-usage=50
ACMEPACKET(system-config)# options +ramdrv-log-min-check=35
ACMEPACKET(system-config)# options +ramdrv-min-log-check=120
ACMEPACKET(system-config)# options +ramdrv-max-log-free=1500
ACMEPACKET(system-config)# options +ramdrv-log-lifetime=7
```

If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Reboot your Oracle Enterprise Session Border Controller.

Configurable Alarm Thresholds and Traps

The Oracle Enterprise Session Border Controller supports user-configurable threshold crossing alarms. These configurations let you identify system conditions of varying severity which create corresponding alarms of varying severity. You configure an alarm threshold type which indicates the resource to monitor. The available types are:

- cpu — CPU utilization monitored as a percentage of total CPU capacity
- memory — memory utilization monitored as a percentage of total memory available

  Note: When you configure an **alarm-threshold** for **memory** with **severity** set to **critical**, the Oracle Enterprise Session Border Controller will stop processing traffic if that configured value is reached, regardless of how low the value is.
- sessions — license utilization monitored as a percentage of licensed session capacity
- space — remaining disk space (configured in conjunction with the volume parameter - see the Storage Expansion Module Monitoring section of the *Accounting Guide* for more information.)
- deny-allocation — denied entry utilization monitored as a percentage of reserved, denied entries.

For the alarm type you create, the Oracle Enterprise Session Border Controller can monitor for 1 through 3 severity levels as minor, major, and critical. Each of the severities is configured with a corresponding value that triggers that severity. For example the configuration for a CPU alarm that is enacted when CPU usage reaches 50%:

```
alarm-threshold
        type                            cpu
        severity                        minor
        value                           50
```

You may create addition CPU alarms for increasing severities. For example:

```
alarm-threshold
       type                            cpu
       severity                        critical
       value                           90
```

The alarm state is enacted when the resource defined with the type parameter exceeds the value parameter. When the resource drops below the value parameter, the alarm is cleared.

SNMP Traps

When a configured alarm threshold is reached, the Oracle Enterprise Session Border Controller sends an apSysMgmtGroupTrap. This trap contains the resource type and value for the alarm configured in the alarm-threshold configuration element. The trap does not contain information associated with configured severity for that value.

```
apSysMgmtGroupTrap          NOTIFICATION-TYPE
    OBJECTS        { apSysMgmtTrapType, apSysMgmtTrapValue }
    STATUS           current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          exceeds a certain threshold. "
    ::= { apSystemManagementNotifications 1 }
```

When the resource usage retreats below a configured threshold, the Oracle Enterprise Session Border Controller sends an apSysMgmtGroupClearTrap.

```
apSysMgmtGroupClearTrap          NOTIFICATION-TYPE
    OBJECTS        { apSysMgmtTrapType }
    STATUS           current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          returns to within a certain threshold.  This signifies that
          an alarm caused by that monitoring object has been cleared. "
    ::= { apSystemManagementNotifications 2 }
```

The alarm and corresponding traps available through the User Configurable Alarm Thresholds functionality are summarized in the following table.

Alarm	Severity	Cause	Actions
CPU	minor major critical	high CPU usage	apSysMgmtGroupTrap sent with apSysCPUUtil apSysMgmtTrapValue
memory	minor major critical	high memory usage	apSysMgmtGroupTrap sent with apSysMemoryUtil apSysMgmtTrapValue
sessions	minor major critical	high license usage	apSysMgmtGroupTrap sent with apSysLicenseCapacity apSysMgmtTrapValue
space	minor major critical	high HDD usage, per volume	apSysMgmtStorageSpaceAvailThresholdTrap sent with: apSysMgmtSpaceAvailCurrent apSysMgmtSpaceAvailMinorThreshold apSysMgmtSpaceAvailMajorThreshold

Alarm	Severity	Cause	Actions
			apSysMgmtSpaceAvailCriticalThreshold apSysMgmtPartitionPath
deny allocation	minor major critical	high usage of denied ACL entries	apSysMgmtGroupTrap sent with apSysCurrentEndptsDenied apSysMgmtTrapValue

Alarm Thresholds Configuration

To configure alarm thresholds:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type system-config and press Enter.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

4. Type alarm-threshold and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system-config)# alarm-threshold
ACMEPACKET(alarm-threshold)#
```

5. type — Enter the type of resource which this alarm monitors. Valid values include:
   - cpu
   - memory
   - sessions
   - space
   - deny-allocation
6. volume — Enter the logical disk volume this alarm monitors (used only in conjunction when type = space).
7. severity — Set the severity of the threshold. Valid values include:
   - minor
   - major
   - critical
8. value — Enter the value from 1 to 99, indicating the percentage, which when exceeded generates an alarm.
9. Save and activate your configuration.

Alarm Synchronization

Two trap tables in the ap-smgmt.mib record trap information for any condition on the Oracle Enterprise Session Border Controller that triggers an alarm condition. You can poll these two tables from network management systems, OSS applications, and the Session Delivery Manager to view the fault status on one or more Oracle Enterprise Session Border Controller s.

The two trap tables that support alarm synchronization, and by polling them you can obtain information about the current fault condition on the Oracle Enterprise Session Border Controller . These tables are:

- apSysMgmtTrapTable—You can poll this table to obtain a summary of the Oracle Enterprise Session Border Controller ’s current fault conditions. The table records multiples of the same trap type that have occurred within a second of one another and have different information. Each table entry contains the following:
  - Trap identifier
  - System time (synchronized with an NTP server)
  - sysUpTime
  - Instance number
  - Other trap information for this trap identifier
- apSysMgmtTrapInformationTable—You can poll this table to obtain further details about the traps recorded in the apSysMgmtTrapTable table. The following information appears:
  - Data index
  - Data type
  - Data length
  - The data itself (in octets)

Trap tables do not record information about alarm severity.

The apSysMgmtTrapTable can hold up to 1000 entries, and you can configure the number of days these entries stay in the table for a maximum of seven days. If you set this parameter to 0 days, the feature is disabled. And if you change the setting to 0 days from a greater value, then the Oracle Enterprise Session Border Controller purges the tables.

Caveats

Note that the Oracle Enterprise Session Border Controller does not replicate alarm synchronization table data across HA nodes. That is, each Oracle Enterprise Session Border Controller in an HA node maintains its own tables.

Alarm Synchronization Configuration

You turn on alarm synchronization in the system configuration.

To use alarm synchronization:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system-config and press Enter.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

4. trap-event-lifetime—To enable alarm synchronization—and cause the Oracle Enterprise Session Border Controller to record trap information in the apSysMgmtTrapTable and the apSysMgmtTrapInformationTable—set this parameter to the number of days you want to keep the information. Leaving this parameter set to 0 (default) turns alarm synchronization off, and you can keep information in the tables for up to 7 days. 7 is the maximum value for this parameter.

Accounting Configuration

The Oracle Enterprise Session Border Controller offers support for RADIUS, an accounting, authentication, and authorization (AAA) system. In general, RADIUS servers are responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver service to the user.

You can configure your Oracle Enterprise Session Border Controller to send call accounting information to one or more RADIUS servers. This information can help you to see usage and QoS metrics, monitor traffic, and even troubleshoot your system.

This guide contains all RADIUS information, as well as information about:

- Accounting for SIP and H.323
- Local CDR storage on the Oracle Enterprise Session Border Controller , including CSV file format settings
- The ability to send CDRs via FTP to a RADIUS sever (the FTP push feature)
- Per-realm accounting control
- Configurable intermediate period
- RADIUS CDR redundancy
- RADIUS CDR content control

Stream Control Transfer Protocol Overview

The Stream Control Transmission Protocol (SCTP) was originally designed by the Signaling Transport (SIGTRAN) group of IETF for Signalling System 7 (SS7) transport over IP-based networks. It is a reliable transport protocol operating on top of an unreliable connectionless service, such as IP. It provides acknowledged, error-free, non-duplicated transfer of messages through the use of checksums, sequence numbers, and selective retransmission mechanism.

SCTP is designed to allow applications, represented as endpoints, communicate in a reliable manner, and so is similar to TCP. In fact, it has inherited much of its behavior from TCP, such as association (an SCTP peer-to-peer connection) setup, congestion control and packet-loss detection algorithms. Data delivery, however, is significantly different. SCTP delivers discrete application messages within multiple logical streams within the context of a single association. This approach to data delivery is more flexible than the single byte-stream used by TCP, as messages can be ordered, unordered or even unreliable within the same association.

SCTP Packets

SCTP packets consist of a common header and one or more chunks, each of which serves a specific purpose.

- DATA chunk — carries user data
- INIT chunk — initiates an association between SCTP endpoints
- INIT ACK chunk — acknowledges association establishment
- SACK chunk — acknowledges received DATA chunks and informs the peer endpoint of gaps in the received subsequences of DATA chunks
- HEARTBEAT chunk — tests the reachability of an SCTP endpoint
- HEARTBEAT ACK chunk — acknowledges reception of a HEARTBEAT chunk
- ABORT chunk — forces an immediate close of an association
- SHUTDOWN chunk — initiates a graceful close of an association
- SHUTDOWN ACK chunk — acknowledges reception of a SHUTDOWN chunk
- ERROR chunk — reports various error conditions
- COOKIE ECHO chunk — used during the association establishment process
- COOKIE ACK chunk — acknowledges reception of a COOKIE ECHO chunk
- SHUTDOWN COMPLETE chunk — completes a graceful association close

SCTP Terminology

This section defines some terms commonly found in SCTP standards and documentation.

SCTP Association

is a connection between SCTP endpoints. An SCTP association is uniquely identified by the transport addresses used by the endpoints in the association. An SCTP association can be represented as a pair of SCTP endpoints, for example, assoc = { [IPv4Addr : PORT1], [IPv4Addr1, IPv4Addr2: PORT2]}.

Only one association can be established between any two SCTP endpoints.

SCTP Endpoint

is a sender or receiver of SCTP packets. An SCTP endpoint may have one or more IP address but it always has one and only one SCTP port number. An SCTP endpoint can be represented as a list of SCTP transport addresses with the same port, for example, endpoint = [IPv6Addr, IPv6Addr: PORT].

An SCTP endpoint may have multiple associations.

SCTP Path

is the route taken by the SCTP packets sent by one SCTP endpoint to a specific destination transport address or its peer SCTP endpoint. Sending to different destination transport addresses does not necessarily guarantee separate routes.

SCTP Primary Path

is the default destination source address, the IPv4 or IPv6 address of the association initiator. For retransmissions however, another active path may be selected, if one is available.

SCTP Stream

is a unidirectional logical channel established between two associated SCTP endpoints. SCTP distinguishes different streams of messages within one SCTP association. SCTP makes no correlation between an inbound and outbound stream.

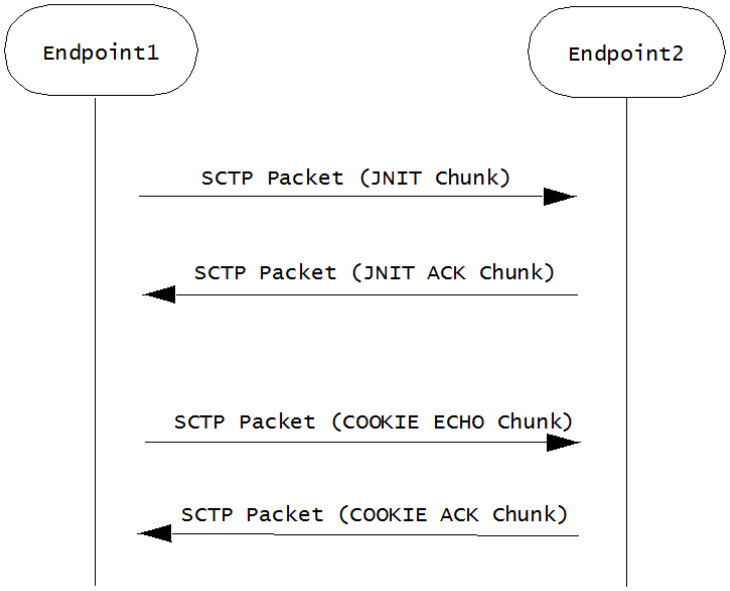
SCTP Transport Address

is the combination of an SCTP port and an IP address. For the current release, the IP address portion of an SCTP Transport Address must be a routable, unicast IPv4 or IPv6 address.

An SCTP transport address binds to a single SCTP endpoint.

SCTP Message Flow

Before peer SCTP users (commonly referred to as endpoints) can send data to each other, an association (an SCTP connection) must be established between the endpoints. During the association establishment process a cookie mechanism is employed to provide protection against security attacks. The following figure shows a sample SCTP association establishment message flow.



Endpoint1 initiates the association by sending Endpoint2 an SCTP packet that contains an INIT chunk, which can include one or more IP addresses used by the initiating endpoint. Endpoint2 acknowledges the initiation of an SCTP association with an SCTP packet that contains an INIT_ACK chunk. This chunk can also include one or more IP addresses at used by the responding endpoint.

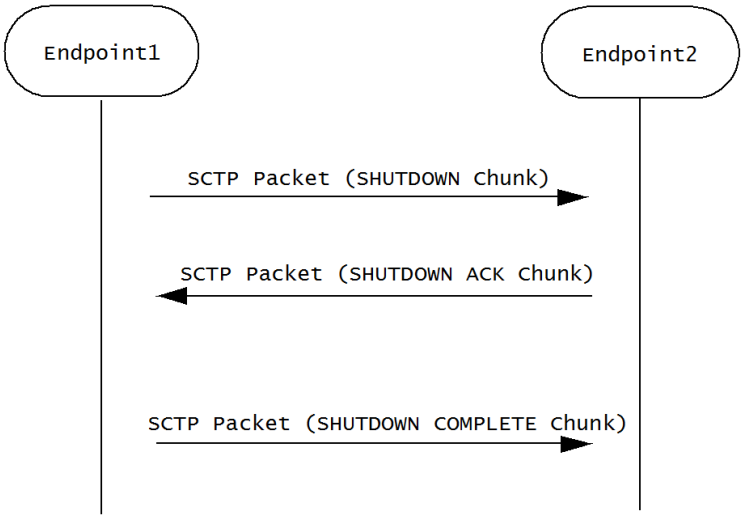
Both the INIT chuck (issued by the initiator) and INIT ACK chunk (issued by the responder) specify the number of outbound streams supported by the association, as well as the maximum inbound streams accepted from the other endpoint.

Association establishment is completed by a COOKIE ECHO/COOKIE ACK exchange that specifies a cookie value used in all subsequent DATA exchanges.

Once an association is successfully established, an SCTP endpoint can send unidirectional data streams using SCTP packets that contain DATA chunks. The recipient endpoint acknowledges with an SCTP packet containing a SACK chunk.

SCTP monitors endpoint reachability by periodically sending SCTP packets that contain HEARTBEAT chunks. The recipient endpoint acknowledges receipt, and confirms availability, with an SCTP packet containing a HEARBEAT ACK chunk.

Either SCTP endpoint can initiate a graceful association close with an SCTP packet that contains a SHUTDOWN chunk. The recipient endpoint acknowledges with an SCTP packet containing a SHUTDOWN ACK chunk. The initiating endpoint concludes the graceful close with an SCTP packet that contains a SHUTDOWN COMPLETE chunk.

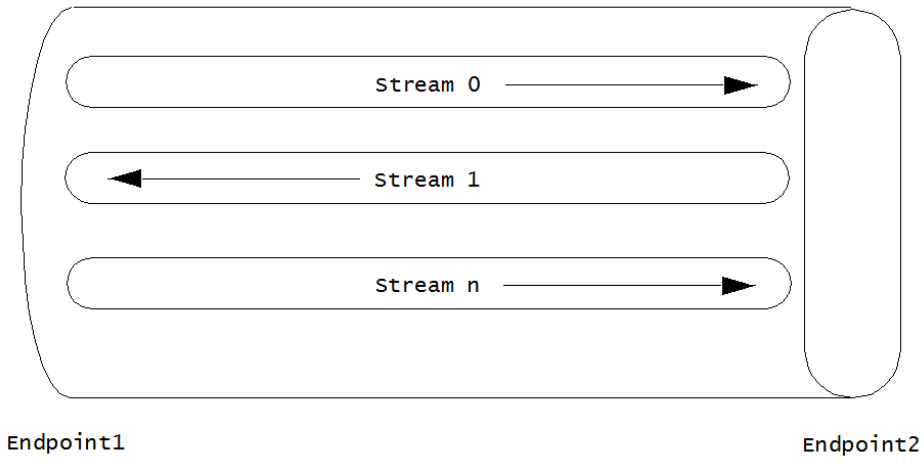


Congestion Control

SCTP congestion control mechanism is similar to that provided by TCP, and includes slow start, congestion avoidance, and fast retransmit. In SCTP, the initial congestion window (cwnd) is set to the double of the maximum transmission unit (MTU) while in TCP, it is usually set to one MTU. In SCTP, cwnd increases based on the number of acknowledged bytes, rather than the number of acknowledgements in TCP. The larger initial cwnd and the more aggressive cwnd adjustment provided by SCTP result in a larger average congestion window and, hence, better throughput performance than TCP.

Multi-Streaming

SCTP supports streams as depicted in the following figure which depicts an SCTP association that supports three streams.



The multiple stream mechanism is designed to solve the head-of-the-line blocking problem of TCP. Therefore, messages from different multiplexed flows do not block one another.

A stream can be thought of as a sub-layer between the transport layer and the upper layer. SCTP supports multiple logical streams to improve data transmission throughput. As shown in the above figure, SCTP allows multiple unidirectional streams within an association. This multiplexing/de-multiplexing capability is called multi-streaming and it is achieved by introducing a field called Stream Identifier contained in every DATA chunk) that is used to differentiate segments in different streams.

SIP transactions are mapped into SCTP streams as described in Section 5.1 of RFC 4168. In what it describes as the simplest way, the RFC suggests (keyword SHOULD) that all SIP messages be transmitted via Stream 0 with the U bit set to 1.

On the transmit side, the current SCTP implementation follows the RFC 4168 recommendation. On the receiving side, a SIP entity must be prepared to receive SIP messages over any stream.

Delivery Modes

SCTP supports two delivery modes, ordered and unordered . Delivery mode is specified by the U bit in the DATA chunk header — if the bit is clear (0), ordered delivery is specified; if the bit is set (1), unordered delivery is specified.

Within a stream, an SCTP endpoint must deliver ordered DATA chunks (received with the U bit set to 0) to the upper layer protocol according to the order of their Stream Sequence Number . Like the U bit, the Stream Sequence Number is a field within the DATA chunk header, and serves to identify the chunk's position with the message stream. If DATA chunks arrive out of order of their Stream Sequence Number, the endpoint must delay delivery to the upper layer protocol until they are reordered and complete.

Unordered DATA chunks (received with the U bit set to 1) are processed differently. When an SCTP endpoint receives an unordered DATA chunk, it must bypass the ordering mechanism and immediately deliver the data to the upper layer protocol (after reassembly if the user data is fragmented by the sender). As a consequence, the Stream Sequence Number field in an unordered DATA chunk has no significance. The sender can fill it with arbitrary value, but the receiver must ignore any value in field.

When an endpoint receives a DATA chunk with the U flag set to 1, it must bypass the ordering mechanism and immediately deliver the data to the upper layer (after reassembly if the user data is fragmented by the data sender).

Unordered delivery provides an effective way of transmitting out-of-band data in a given stream. Note also, a stream can be used as an unordered stream by simply setting the U bit to 1 in all DATA chunks sent through that stream.

Multi-Homing

Call control applications for carrier-grade service require highly reliable communication with no single point of failure. SCTP can assist carriers with its multi-homing capabilities. By providing different paths through the network over separate and diverse means, the goal of no single point of failure is more easily attained.

SCTP built-in support for multi-homed hosts allows a single SCTP association to run across multiple links or paths, hence achieving link/path redundancy. With this capability, and SCTP association can be made to achieve fast failover from one link/path to another with little interruption to the data transfer service.

Multi-homing enables an SCTP host to establish an association with another SCTP host over multiple interfaces identified by different IP addresses. With specific regard to the Oracle Enterprise Session Border Controller these IP addresses need not be assigned to the same physical interface, or to the same physical Network Interface Unit.

If the SCTP nodes and the according IP network are configured in such a way that traffic from one node to another travels on physically different paths if different destination IP address are used, associations become tolerant against physical network failures and other problems of that kind.

An endpoint can choose an optimal or suitable path towards a multi-homed destination. This capability increases fault tolerance. When one of the paths fails, SCTP can still choose another path to replace the previous one. Data is always sent over the primary path if it is available. If the primary path becomes unreachable, data is migrated to a different, affiliated address — thus providing a level of fault tolerance. Network failures that render one interface of a server unavailable do not necessarily result in service loss. In order to achieve real fault resilient communication between two SCTP endpoints, the maximization of the diversity of the round-trip data paths between the two endpoints is encouraged.

Multi-Homing and Path Diversity

As previously explained, when a peer is multi-homed, SCTP can automatically switch the subsequent data transmission to an alternative address. However, using multi-homed endpoints with SCTP does not automatically guarantee resilient communications. One must also design the intervening network(s) properly.

To achieve fault resilient communication between two SCTP endpoints, one of the keys is to maximize the diversity of the round-trip data paths between the two endpoints. Under an ideal situation, one can make the assumption that every destination address of the peer will result in a different, separate path towards the peer. Whether this can be achieved in practice depends entirely on a combination of factors that include path diversity, multiple connectivity, and the routing protocols that glue the network together. In a normally designed network, the paths may not be diverse, but there may be multiple connectivity between two hosts so that a single link failure will not fail an association.

In an ideal arrangement, if the data transport to one of the destination addresses (which corresponds to one particular path) fails, the data sender can migrate the data traffic to other remaining destination address(es) (that is, other paths) within the SCTP association.

Monitoring Failure Detection and Recovery

When an SCTP association is established, a single destination address is selected as the primary destination address and all new data is sent to that primary address by default. This means that the behavior of a multi-homed SCTP association when there are no network losses is similar to behavior of a TCP connection. Alternate, or secondary, destination addresses are only used for redundancy purposes, either to retransmit lost packets or when the primary destination address cannot be reached.

A failover to an alternate destination is performed when the SCTP sender cannot elicit an acknowledgement — either a SACK for a DATA chunk, or a HEARTBEAT ACK for a HEARTBEAT chunk — for a configurable consecutive number of transmissions. The SCTP sender maintains an error-counter is maintained for each destination address and if this counter exceeds a threshold (normally six), the address is marked as inactive, and taken out of service. If the primary destination address is marked as inactive, all data is then switched to a secondary address to complete the failover.

If no data has been sent to an address for a specified time, that endpoint is considered to be idle and a HEARTBEAT packet is transmitted to it. The endpoint is expected to respond to the HEARTBEAT immediately with a HEARTBEAT ACK. As well as monitoring the status of destination addresses, the HEARTBEAT is used to obtain RTT measurements on idle paths. The primary address becomes active again if it responds to a heartbeat.

The number of events where heartbeats were not acknowledged within a certain time, or retransmission events occurred is counted on a per association basis, and if a certain limit is exceeded, the peer endpoint is considered unreachable, and the association is closed.

The threshold for detecting an endpoint failure and the threshold for detecting a failure of a specific IP addresses of the endpoint are independent of each other. Each parameter can be separately configured by the SCTP user. Careless configuration of these protocol parameters can lead the association onto the dormant state in which all the destination addresses of the peer are found unreachable while the peer still remains in the reachable state. This is because the overall retransmission counter for the peer is still below the set threshold for detecting the peer failure.

Configuring SCTP Support for SIP

RFC 4168, *The Stream Control Transfer Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)*, specifies the requirements for SCTP usage as a layer 4 transport for SIP. Use the following steps to:

- configure SCTP as the layer 4 transport for a SIP interface
- create an SCTP-based SIP port
- associate physical interfaces/network interfaces with SIP realms
- identify adjacent SIP servers that are accessible via SCTP
- set SCTP timers and counters (optional)

Configuring an SCTP SIP Port

SIP ports are created as part of the SIP Interface configuration process.

1. From superuser mode, use the following command sequence to access sip-port configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)# sip-ports
ACMEPACKET(sip-port)#
```

2. Use the address parameter to provide the IPv4 or IPv6 address of the network interface that supports the SIP port.

   This is the primary address of a the local multi-homed SCTP endpoint.

```
ACMEPACKET(sip-port)# address 172.16.10.76
ACMEPACKET(sip-port)#
```

3. Retain the default value, 5060 (the well-known SIP port) for the port parameter.

```
ACMEPACKET(sip-port)# port 5060
ACMEPACKET(sip-port)#
```

4. Use the transport-protocol parameter to identify the layer 4 protocol.

   Supported values are UDP, TCP, TLS, and SCTP.

   Select SCTP.

```
ACMEPACKET(sip-port)# transport-protocol sctp
ACMEPACKET(sip-port)#
```

5. Use the multi-homed-addrs parameter to specify one or more local secondary addresses of the SCTP endpoint.

   Multi-homed addresses must be of the same type (IPv4 or IPv6) as that specified by the address parameter. Like the address parameter, these addresses identify SD physical interfaces.

   To specify multiple addresses, bracket an address list with parentheses.

```
ACMEPACKET(sip-port)# multi-homed-addrs 182.16.10.76
ACMEPACKET(sip-port)#
ACMEPACKET(sip-port)# multi-homed-addrs (182.16.10.76 192.16.10.76
196.15.32.108)
ACMEPACKET(sip-port)#
```

6. Remaining parameters can be safely ignored.
7. Use done, exit, and verify-config to complete configuration of this SCTP-based SIP port.

```
ACMEPACKET(sip-port)# done
ACMEPACKET(sip-interface)# exit
ACMEPACKET(session-router)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Configuring the Realm

After configuring a SIP port which identifies primary and secondary multi-homed transport addresses, you identify the network interfaces that support the primary address and secondary addresses to the realm assigned during SIP Interface configuration.

1. From superuser mode, use the following command sequence to access realm-config configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

2. Use the select command to access the target realm.
3. Use the network-interfaces command to identify the network interfaces that support the SCTP primary and secondary addresses.

   Network interfaces are identified by their name.

Enter a list of network interface names using parentheses as list brackets. The order of interface names is not significant.

```
ACMEPACKET(realm-config)# network-interfaces (mo1 M10)
ACMEPACKET(realm-config)#
```

4. Use done, exit, and verify-config to complete realm configuration.

```
ACMEPACKET(realm-config)# done
ACMEPACKET(media-manager)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Configuring Session Agents

After configuring the realm, you identify adjacent SIP servers who will be accessed via the SCTP protocol.

1. From superuser mode, use the following command sequence to access session-agent configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

2. Use the select command to access the target session-agent.
3. Use the transport-method parameter to select the layer 4 transport protocol.

   Select staticSCTP for SCTP transport

```
ACMEPACKET(session-agent)# transport-method staticSCTP
ACMEPACKET(session-agent)#
```

4. Set the reuse-connections parameter to none.

   Select staticSCTP for SCTP transport

```
ACMEPACKET(session-agent)# reuse-connections none
ACMEPACKET(session-agent)#
```

5. Use done, exit, and verify-config to complete session agent configuration.

```
ACMEPACKET(session-agent)# done
ACMEPACKET(session-router)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

6. Repeat Steps 1 through 5 as necessary to configure additional session agents who will be accessed via SCTP transport.

Setting SCTP Timers and Counters

Setting SCTP timers and counters is optional. All configurable timers and counters provide default values that conform to recommended values as specified in RFC 4960, Stream Control Transmission Protocol.

Management of Retransmission Timer, section 6.3 of RFC 4960 describes the calculation of a Retransmission Timeout (RTO) by the SCTP process. This calculation involves three SCTP protocol parameters: RTO.Initial, RTO.Min, and RTO.Max. Suggested SCTP Protocol Parameter Values section 15 of RFC 4960 lists recommended values for these parameters.

The following shows the equivalence of recommended values and ACLI defaults.

RTO.Initial = 3 seconds sctp-rto-initial = 3000 ms (default value)

RTO.Min = 1 second sctp-rto-min = 1000 ms (default value)

RTO.Max = 60 seconds sctp-rto-max = 60000 ms (default value)

Path Heartbeat, section 8.3 of RFC 4960 describes the calculation of a Heartbeat Interval by the SCTP process. This calculation involves the current calculated RTO and a single SCTP protocol parameter — HB.Interval.

The following shows the equivalence of recommended the value and ACLI default.

HB.Interval = 30 seconds sctp-hb-interval = 3000 ms (default value)

Acknowledgement on Reception of DATA Chunks, section 6.2 of RFC 4960 describes requirements for the timely processing and acknowledgement of DATA chunks. This section requires that received DATA chunks must be acknowledged within 500 milliseconds, and recommends that DATA chunks should be acknowledged with 200 milliseconds. The interval between DATA chunk reception and acknowledgement is specific by the ACLI sctp-sack-timeout parameter, which provides a default value of 200 milliseconds and a maximum value of 500 milliseconds.

Transmission of DATA Chunks, section 6.1 of RFC 4960 describes requirements for the transmission of DATA chunks. To avoid network congestion the RFC recommends a limitation on the volume of data transmitted at one time. The limitation is expressed in terms of DATA chunks, not in terms of SCTP packets.

The maximum number of DATA chunks that can be transmitted at one time is specified by the ACLI sctp-max-burst parameter, which provides a default value of 4 chunks, the limit recommended by the RFC.

Setting the RTO

An SCTP endpoint uses a retransmission timer to ensure data delivery in the absence of any feedback from its peer. RFC 4960 refers to the timer itself as T3-rtx and to the timer duration as RTO (retransmission timeout).

When an endpoint's peer is multi-homed, the endpoint calculates a separate RTO for each IP address affiliated with the peer. The calculation of RTO in SCTP is similar to the way TCP calculates its retransmission timer. RTO fluctuates over time in response to actual network conditions. To calculate the current RTO, an endpoint maintains two state variables per destination IP address — the SRTT (smoothed round-trip time) variable, and the RTTVAR (round-trip time variation) variable.

Use the following procedure to assign values used in RTO calculation.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-rto-initial parameter to assign an initial timer duration.

   Allowable values are integers within the range 0 through 4294967295 that specify the initial duration in milliseconds. In the absence of an explicitly configured integer value, sctp-rto-initial defaults to 3000 milliseconds (3 seconds, the recommended default value from RFC 4960).

   As described in Section 6.3 of RFC 4960, the value specified by sctp-rto-initial is assigned to the SCTP protocol parameter RTO.Initial, which provides a default RTO until actual calculations have derived a fluctuating duration based on network usage. The value specified by the sctp-rto-initial parameter seeds these calculations.

```
ACMEPACKET(network-parameters)# sctp-rto-initial 3000
ACMEPACKET(network-parameters)#
```

3. Use the sctp-rto-min and sctp-rto-max parameters to assign an RTO floor and ceiling.

   Allowable values are integers within the range 0 through 4294967295 that specify the minimum and maximum durations in milliseconds. In the absence of an explicitly configured integer value, sctp-rto-min defaults to 1000 ms (1 second, the recommended default value from RFC 4960), and sctp-rto-max defaults to 60000 ms (60 seconds, the recommended default value from RFC 4960.)

   As described in Section 6.3 of RFC 4960, the values specified by sctp-rto-min and sctp-rto-max are assigned to the SCTP protocol parameters, RTO.min and RTO.max that limit RTO calculations. If a calculated RTO duration

is less than RTO.min, the parameter value is used instead of the calculated value; likewise, if a calculated RTO duration is greater than RTO.max, the parameter value is used instead of the calculated value.

```
ACMEPACKET(network-parameters)# sctp-rto-min 1000
ACMEPACKET(network-parameters)# sctp-rto-max 60000
ACMEPACKET(network-parameters)#
```

4. Use done, exit, and verify-config to complete RTO configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting the Heartbeat Interval

Both single-homed and multi-homed SCTP endpoints test the reachability of associates by sending periodic HEARTBEAT chunks to UNCONFIRMED or idle transport addresses.

Use the following procedure to assign values used in Heartbeat Interval calculation.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-hb-interval parameter to assign an initial Heartbeat Interval duration.

   Allowable values are integers within the range 0 through 4294967295 that specify the initial Heartbeat Interval in milliseconds. In the absence of an explicitly configured integer value, sctp-hb-interval defaults to 30000 milliseconds (30 seconds, the recommended default value from RFC 4960).

   As described in Section 8.3 of RFC 4960, the value specified by sctp-hb-interval is assigned to the SCTP protocol parameter HB.Interval, which provides a default interval until actual calculations have derived a fluctuating interval based on network usage. The value specified by the sctp-hb-interval parameter is used during these calculations.

```
ACMEPACKET(network-parameters)# sctp-hb-interval 30000
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete Heartbeat Interval configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET #
```

Setting the SACK Delay Timer

An SCTP Selective Acknowledgement (SACK) is sent to the peer endpoint to acknowledge received DATA chunks and to inform the peer endpoint of gaps in the received subsequences of DATA chunks. Section 6.2 of RFC 4960 sets a specific requirement for a SACK Delay timer that specifies the maximum interval between the reception of an SCTP packet containing one or more DATA chunks and the transmission of a SACK to the packet originator.

Use the following procedure to set the SACK Delay timer.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-sack-timeout parameter to assign a value to the SACK Delay timer.

   Allowable values are integers within the range 0 through 500 which specify the maximum delay (in milliseconds) between reception of a SCTP packet containing one or more Data chunks and the transmission of a SACK to the packet source. The value 0 indicates that a SACK is generated immediately upon DATA chunk reception

   In the absence of an explicitly configured integer value, sctp-sack-timeout defaults to 200 ms (the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-sack-timeout 200
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete configuration of the SACK Delay timer.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Limiting DATA Bursts

Section 6.1 of RFC 4960 describes the SCTP protocol parameter, Max.Burst, used to limit the number of DATA chunks that are transmitted at one time.

Use the following procedure to assign a value to the SCTP protocol parameter, Max.Burst.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-max-burst parameter to assign a value to the SCTP protocol parameter, Max.Burst.

   Allowable values are integers within the range 0 through 4294967295 that specify the maximum number of DATA chunks that will be sent at one time. In the absence of an explicitly configured integer value, sctp-max-burst defaults to 4 (DATA chunks, the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-max-burst 4
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete configuration of DATA burst limitations.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting Endpoint Failure Detection

As described in Monitoring, Failure Detection and Recovery, a single-homed SCTP endpoint maintains a count of the total number of consecutive failed (unacknowledged) retransmissions to its peer. Likewise, a multi-homed SCTP endpoint maintains a series of similar, dedicated counts for all of its destination transport addresses. If the value of these counts exceeds the limit indicated by the SCTP protocol parameter Association.Max.Retrans, the endpoint

considers the peer unreachable and stops transmitting any additional data to it, causing the association to enter the CLOSED state.

The endpoint resets the counter when (1) a DATA chunk sent to that peer endpoint is acknowledged by a SACK, or (2) a HEARTBEAT ACK is received from the peer endpoint.

Use the following procedure to configure endpoint failure detection.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-assoc-max-retrans to assign a value to the SCTP protocol parameter Association.Max.Retrans.

   Allowable values are integers within the range 0 through 4294967295 which specify the maximum number of transmission requests. In the absence of an explicitly configured integer value, sctp-assoc-max-retrans defaults to 10 (transmission re-tries, the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-assoc-max-retrans 10
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete endpoint failure detection configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting Path Failure Detection

As described in Monitoring, Failure Detection and Recovery, when its peer endpoint is multi-homed, an SCTP endpoint maintains a count for each of the peer's destination transport addresses.

Each time the T3-rtx timer expires on any address, or when a HEARTBEAT sent to an idle address is not acknowledged within an RTO, the count for that specific address is incremented. If the value of a specific address count exceeds the SCTP protocol parameter Path.Max.Retrans, the endpoint marks that destination transport address as inactive.

The endpoint resets the counter when (1) a DATA chunk sent to that peer endpoint is acknowledged by a SACK, or (2) a HEARTBEAT ACK is received from the peer endpoint.

When the primary path is marked inactive (due to excessive retransmissions, for instance), the sender can automatically transmit new packets to an alternate destination address if one exists and is active. If more than one alternate address is active when the primary path is marked inactive, a single transport address is chosen and used as the new destination transport address.

Use the following procedure to configure path failure detection.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-path-max-retrans parameter to assign a value to the SCTP protocol parameter Path.Max.Retrans.

   Allowable values are integers within the range 0 through 4294967295 that specify the maximum number of RTOs and unacknowledged HEARTBEATS. In the absence of an explicitly configured integer value, sctp-path-max-retrans defaults to 5 (RTO and/or HEARTBEAT errors per transport address, the recommended default value from RFC 4960).

When configuring endpoint and path failure detection, ensure that the value of the sctp-assoc-max-retrans parameter is smaller than the sum of the sctp-path-max-retrans values for all the remote peer's destination addresses. Otherwise, all the destination addresses can become inactive (unable to receive traffic) while the endpoint still considers the peer endpoint reachable.

```
ACMEPACKET(network-parameters)# sctp-path-max-retrans 5
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete path failure detection configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Specifying the Delivery Mode

As described in Delivery Modes, SCTP support two delivery modes, ordered and unordered.

1. From superuser mode, use the following command sequence to access network-parameters configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the sctp-send-mode parameter to select the preferred delivery mode.

   Choose ordered or unordered.

```
ACMEPACKET(network-parameters)# sctp-send-mode unordered
ACMEPACKET(network-parameters)#
```

3. Use done, exit, and verify-config to complete delivery mode configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET #
```

Example Configurations

The following ACLI command sequences summarize required SCTP port configuration, and the configuration of required supporting elements.

- PHY interfaces
- Network interfaces
- SIP ports
- realms
- session agents

Sequences show only configuration parameters essential for SCTP operations; other parameters can retain default values, or assigned other values specific to local network requirements.

Phy Interface Configuration

The first ACLI command sequence configures a physical interface named m10, that will support an SCTP primary address; the second sequence configures an interface named m01 that will support a secondary SCTP address.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# operation-type media
ACMEPACKET(phy-interface)# port 0
ACMEPACKET(phy-interface)# slot 1
ACMEPACKET(phy-interface)# name m10
ACMEPACKET(phy-interface)#
...
...
...
ACMEPACKET(phy-interface)#
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# operation-type media
ACMEPACKET(phy-interface)# port 1
ACMEPACKET(phy-interface)# slot 0
ACMEPACKET(phy-interface)# name m01
ACMEPACKET(phy-interface)#
...
...
...
ACMEPACKET(phy-interface)#
```

Network Interface Configuration

These ACLI command sequences configure two network interfaces. The first sequence configures a network interface named m10, thus associating the network interface with the physical interface of the same name. The ACLI ip-address command assigns the IPv4 address 172.16.10.76 to the network interface. In a similar fashion, the second command sequence associates the m01 network and physical interfaces, and assigns an IPv4 address of 182.16.10.76.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# name m10
ACMEPACKET(network-interface)# ip-address 172.16.10.76
...
...
...
ACMEPACKET(network-interface)#
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# name m01
ACMEPACKET(network-interface)# ip-address 182.16.10.76
...
...
...
ACMEPACKET(network-interface)#
```

SIP Port Configuration

This ACLI command sequence configures a SIP port for SCTP operations. It specifies the use of SCTP as the transport layer protocol, and assigns the existing network interface address, 172.16.10.76, as the SCTP primary address. Additionally, it identifies three other existing network addresses (182.16.10.76, 192.16.10.76, and 196.15.32.108) as SCTP secondary addresses.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)# sip-ports
ACMEPACKET(sip-port)# address 172.16.10.76
ACMEPACKET(sip-port)# transport-protocol sctp
```

```
ACMEPACKET(sip-port)# multi-homed-addrs (182.16.10.76 192.16.10.76
196.15.32.108)
...
...
...
ACMEPACKET(sip-port)#
```

Realm Configuration

These ACLI command sequences configure a realm for SCTP operations. The first ACLI sequence assigns a named realm, in this example core-172, to a SIP interface during the interface configuration process. The second sequence accesses the target realm and uses the network-interfaces command to associate the named SCTP network interfaces with the realm.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)# realm-id core-172
...
...
...
ACMEPACKET(sip-interface)#
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)# select
identifier: core-172
1. core-172 ...
selection: 1
ACMEPACKET(realm-config)# network-interfaces (m01 m10 ...)
...
...
...
ACMEPACKET(ream-config)#
```

Session Agent Configuration

The final ACLI command sequence enables an SCTP-based transport connection between the Acme Packet 4500 and an adjacent network element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)# select
<hostname>: core-172S1
1. core-172S1 ...
selection: 1
ACMEPACKET(session-agent)#
ACMEPACKET(session-agent)# transport-method staticSCTP
ACMEPACKET(session-agent)# reuse-connections none
...
...
...
ACMEPACKET(session-agent)#
```

About the Acme Packet 3800 and Acme Packet 4500 and IPv6

The Acme Packet 3800 and the Acme Packet 4500 support IPv6. Ideally, IPv6 support would be a simple matter of configuring IP addresses of the version type you want in the configurations where you want them. While this is true for some configuration areas, in others you will need to take care with—for example—the format of your IPv6 address entries or where parameters must be configured with IP addresses of the same version type.

This section explains the changes to the ACLI of which you need to be aware as you start to use IPv6 on the Acme Packet 3800 or Acme Packet 4500. Note that not all configurations and their parameters are available for IPv6 use.

Licensing

IPv6 is a licensed feature on the Acme Packet 3800 and on the Acme Packet 4500. If you want to add this license to a system, contact your Oracle sales engineer for information related to the license.

You do not need to take action if you are working with a new system with which the IPv6 license was purchased.

Updated ACLI Help Text

As you complete configuration work and perform monitoring tasks on your system, you might note that there have been changes to the help text to reflect the addition of IPv6 support. These changes are minor, but nonetheless reflect feature support.

In the ACLI that supports only IPv4, there are many references to that version as the accepted value for a configuration parameter or other IPv4-specific languages. For IPv6 support, these references have been edited. For example, rather than providing help that refers specifically to IPv4 addresses when explaining what values are accepted in an ACLI configuration parameter, you will now see an <ipAddr> note.

IPv6 Address Configuration

IPv6 can be a licensed feature on the Oracle Enterprise Session Border Controller. If you want to add this license to a system, then contact your Acme Packet sales engineering for information related to the license. Once you have the license information, refer to the Getting Started chapter for instructions about how to add a license.

You do not need to take action if you are working with a new system with which the IPv6 license was purchased.

Note: For ACLI parameters that support only IPv4, there are many references to that version as the accepted value for a configuration parameter or other IPv4-specific languages. For IPv6 support, these references have been edited. For example, rather than providing help that refers specifically to IPv4 addresses when explaining what values are accepted in an ACLI configuration parameter, you will now see an <ipAddr> note.

This section calls out the configurations and parameters for which you can enter IPv6 addresses. In this first IPv6 implementation, the complete range of system configurations and their parameters are available for IPv6 use.

The Oracle Enterprise Session Border Controller follows RFC 3513 its definition of IPv6 address representations. Quoting from that RFC, these are the two forms supported:

- The preferred form is x:x:x:x:x:x:x:x, where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address. Examples:

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

1080:0:0:0:8:800:200C:417A

Note that it is not necessary to write the leading zeros in an individual field, but there must be at least one numeral in every field (except for the case described in 2.).

- Due to some methods of allocating certain styles of IPv6 addresses, it will be common for addresses to contain long strings of zero bits. In order to make writing addresses containing zero bits easier a special syntax is available to compress the zeros. The use of "::" indicates one or more groups of 16 bits of zeros. The "::" can only appear once in an address. The "::" can also be used to compress leading or trailing zeros in an address. For example, the following addresses: 1080:0:0:0:8:800:200C:417A a unicast address FF01:0:0:0:0:0:0:101 a multicast address

  0:0:0:0:0:0:0:1 the loopback address

  0:0:0:0:0:0:0:0 the unspecified addresses

  may be represented as:

  1080::8:800:200C:417A a unicast address

FF01::101 a multicast address

::1 the loopback address

:: the unspecified addresses

Access Control

These are the IPv6-enabled parameters in the access-control configuration.

Parameter	Entry Format
source-address	<ip-address>[/<num-bits>][:<port>[/<port-bits>]]
destination-address	<ip-address>[/<num-bits>][:<port>[/<port-bits>]]

Host Route

These are the IPv6-enabled parameters in the host-route configuration.

Parameter	Entry Format
dest-network	<ipv4> | <ipv6>
netmask	<ipv4> | <ipv6>
gateway	<ipv4> | <ipv6>

Local Policy

These are the IPv6-enabled parameters in the local-policy configuration.

Parameter	Entry Format
from-address	<ipv4> | <ipv6> | POTS Number, E.164 Number, hostname, wildcard
to-address	<ipv4> | <ipv6> | POTS Number, E.164 Number, hostname, wildcard

Network Interface

These are the IPv6-enabled parameters in the network-interface configuration.

Parameter	Entry Format
hostname	<ipv4> | <ipv6> | hostname
ip-address	<ipv4> | <ipv6>
pri-utility-addr	<ipv4> | <ipv6>
sec-utility-addr	<ipv4> | <ipv6>
netmask	<ipv4> | <ipv6>
gateway	<ipv4> | <ipv6>
sec-gateway	<ipv4> | <ipv6>
dns-ip-primary	<ipv4> | <ipv6>
dns-ip-backup1	<ipv4> | <ipv6>
dns-ip-backup2	<ipv4> | <ipv6>
add-hip-ip	<ipv4> | <ipv6>

Parameter	Entry Format
remove-hip-ip	<ipv4> | <ipv6>
add-icmp-ip	<ipv4> | <ipv6>
remove-icmp-ip	<ipv4> | <ipv6>

Realm Configuration

These are the IPv6-enabled parameters in the realm-config.

Parameter	Entry Format
addr-prefix	[<ipv4> | <ipv6>]/prefix

Session Agent

These are the IPv6-enabled parameters in the session-agent configuration.

Parameter	Entry Format
hostname	<ipv4> | <ipv6>
ip-address	<ipv4> | <ipv6>

SIP Configuration

These are the IPv6-enabled parameters in the session-config.

Parameter	Entry Format
registrar-host	<ipv4> | <ipv6> | hostname | *

SIP Interface SIP Ports

These are the IPv6-enabled parameters in the sip-interface>sip-ports configuration.

Parameter	Entry Format
address	<ipv4> | <ipv6>

Steering Pool

These are the IPv6-enabled parameters in the steering-pool configuration.

Parameter	Entry Format
ip-address	<ipv4> | <ipv6>

System Configuration

These are the IPv6-enabled parameters in the system-config.

Parameter	Entry Format
default-v6-gateway	<ipv6>

IPv6 Default Gateway

In the system configuration, you configure a default gateway—a parameter that now has its own IPv6 equivalent.

To configure an IPv6 default gateway:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system-config and press Enter.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```

4. default-v6-gateway—Set the IPv6 default gateway for this Oracle Enterprise Session Border Controller. This is the IPv6 egress gateway for traffic without an explicit destination. The application of your Oracle Enterprise Session Border Controller determines the configuration of this parameter.
5. Save your work.

Network Interfaces and IPv6

You set many IP addresses in the network interface, one of which is the specific IP address for that network interface and others that are related to different types of management traffic. This section outlines rules you must follow for these entries.

- For the network-interface ip-address parameter, you can set a single IP address. When you are working with an IPv6-enabled system, however, note that all other addresses related to that network-interface IP address must be of the same version.
- Heterogeneous address family configuration is prevented for the dns-ip-primary, dns-ip-backup1, and dns-ip-backup2 parameters.
- For HIP addresses (add-hip-ip), you can use either IPv4 or IPv6 entries.
- For ICMP addresses (add-icmp-ip), you can use either IPv4 or IPv6 entries.
- For Telnet (add-telnet-ip), FTP (add-ftp-ip), and SNMP (add-snmp-ip), you are not allowed to use IPv6; your entries MUST use IPv4.

IPv6 Reassembly and Fragmentation Support

As it does for IPv4, the Oracle Enterprise Session Border Controller supports reassembly and fragmentation for large signaling packets when you enable IPV6 on your system.

The Oracle Enterprise Session Border Controller takes incoming fragments and stores them until it receives the first fragment containing a Layer 4 header. With that header information, the Oracle Enterprise Session Border Controller performs a look-up so it can forward the packets to its application layer. Then the packets are re-assembled at the applications layer. Media fragments, however, are not reassembled and are instead forwarded to the egress interface.

On the egress side, the Oracle Enterprise Session Border Controller takes large signaling messages and encodes it into fragment datagrams before it transmits them.

Note that large SIP INVITE messages should be sent over TCP. If you want to modify that behavior, you can use the SIP interface's option parameter max-udp-length=xx for each SIP interface where you expect to receive large INVITE packets.

Other than enabling IPv6 on your Oracle Enterprise Session Border Controller, there is no configuration for IPv6 reassembly and fragmentation support. It is enabled automatically.

Access Control List Support

The Oracle Enterprise Session Border Controller supports IPv6 for access control lists in two ways:

- For static access control lists that you configure in the access-control configuration, your entries can follow IPv6 form. Further, this configuration supports a prefix that enables wildcarding the source IP address.
- Dynamic ACLs are also supported; the Oracle Enterprise Session Border Controller will create ACLs for offending IPv6 endpoints.

Data Entry

When you set the source-address and destination-address parameters in the access-control configuration, you will use a slightly different format for IPv6 than for IPv4.

For the source-address, your IPv4 entry takes the following format: <ip-address>[/<num-bits>][:<port>[/<port-bits>]]. And for the destination-address, your IPv4 entry takes this format: <ip-address>[:<port>[/<port-bits>]].

Since the colon (:) in the IPv4 format leads to ambiguity in IPv6, your IPv6 entries for these settings must have the address encased in brackets ([]): [7777::11]/64:5000/14.

In addition, IPv6 entries are allowed up to 128 bits for their prefix lengths.

The following is an example access control configuration set up with IPv6 addresses.

```
ACMEPACKET(access-control)# done
access-control
        realm-id                        net7777
        description
        source-address                  7777::11/64:5060/8
        destination-address             8888::11:5060/8
        application-protocol            SIP
        transport-protocol              ALL
        access                          deny
        average-rate-limit              0
        trust-level                     none
        minimum-reserved-bandwidth      0
        invalid-signal-threshold        10
        maximum-signal-threshold        0
        untrusted-signal-threshold      0
        deny-period                     30
```

DNS Support

The Oracle Enterprise Session Border Controller supports the DNS resolution of IPv6 addresses; in other words, it can request the AAAA record type (per RFC 1886) in DNS requests. In addition, the Oracle Enterprise Session Border Controller can make DNS requests over IPv6 transport so that it can operate in networks that host IPv6 DNS servers.

For mixed IPv4-IPv6 networks, the Oracle Enterprise Session Border Controller follows these rules:

- If the realm associated with the name resolution is an IPv6 realm, the Oracle Enterprise Session Border Controller will send the query out using the AAAA record type.
- If the realm associated with the name resolution is an IPv4 realm, the Oracle Enterprise Session Border Controller will send the query out using the A record type.

In addition, heterogeneous address family configuration is prevented for the dns-ip-primary, dns-ip-backup1, and dns-ip-backup2 parameters.

Homogeneous Realms

IPv6 is supported for realms and for nested realms, as long as the parent chain remains within the same address family. If you try to configure realms with mixed IPv4-IPv6 addressing, your system will issue an error message when you try to save your configuration. This check saves you time because you do not have to wait to run a configuration verification (using the ACLI verify-config command) to find possible errors.

Parent-Child Network Interface Mismatch

Your system will issue the following error message if parent-child realms are on different network interfaces that belong to different address families:

```
ERROR: realm-config [child] and parent [net8888] are on network interfaces
that belong to different address families
```

Address Prefix-Network Interface Mismatch

If the address family and the address-prefix you configure for the realm does not match the address family of its network interface, your system will issue the following error message:

```
ERROR: realm-config [child] address prefix and network interface [1:1:0]
belong to different address families
```

RADIUS Support for IPv6

The Oracle Enterprise Session Border Controller's RADIUS support now includes:

- RADIUS CDR generation for SIPv6-SIPv6 and SIPv6-SIPv4 calls
- IPv6-based addresses in RADIUS CDR attributes

This means that for the CDR attributes in existence prior to the introduction of IPv6 to the Acme Packet 4500 are mapped to the type ipaddr, which indicates four-byte field. The sixteen-byte requirement for IPv6 addresses is now supported, and there are a parallel set of attributes with the type ipv6addr. Attributes 155-170 are reserved for the IPv6 addresses.

NAS addresses use the number 95 to specify the NAS-IPV6-Address attribute. And local CDRs now contain IPv6 addresses.

Supporting RADIUS VSAs

The following VSAs have been added to the Oracle RADIUS dictionary to support IPv6.

```
Acme-Flow-In-Src-IPv6_Addr_FS1_F        155     ipv6addr      Acme
Acme-Flow-In-Dst-IPv6_Addr_FS1_F        156     ipv6addr      Acme
Acme-Flow-Out-Src-IPv6_Addr_FS1_F       157     ipv6addr      Acme
Acme-Flow-Out-Dst-IPv6_Addr_FS1_F       158     ipv6addr      Acme
Acme-Flow-In-Src-IPv6_Addr_FS1_R        159     ipv6addr      Acme
Acme-Flow-In-Dst-IPv6_Addr_FS1_R        160     ipv6addr      Acme
Acme-Flow-Out-Src-IPv6_Addr_FS1_R       161     ipv6addr      Acme
Acme-Flow-Out-Dst-IPv6_Addr_FS1_R       162     ipv6addr      Acme
Acme-Flow-In-Src-IPv6_Addr_FS2_F        163     ipv6addr      Acme
Acme-Flow-In-Dst-IPv6_Addr_FS2_F        164     ipv6addr      Acme
Acme-Flow-Out-Src-IPv6_Addr_FS2_F       165     ipv6addr      Acme
Acme-Flow-Out-Dst-IPv6_Addr_FS2_F       166     ipv6addr      Acme
Acme-Flow-In-Src-IPv6_Addr_FS2_R        167     ipv6addr      Acme
Acme-Flow-In-Dst-IPv6_Addr_FS2_R        168     ipv6addr      Acme
Acme-Flow-Out-Src-IPv6_Addr_FS2_R       169     ipv6addr      Acme
Acme-Flow-Out-Dst-IPv6_Addr_FS2_R       170     ipv6addr      Acme
```

7

Realms and Nested Realms

This chapter explains how to configure realms and nested realms, and specialized media-related features.

A realm is a logical definition of a network or groups of networks made up in part by devices that provide real-time communication sessions comprised of signaling messages and possibly media flows. These network devices might be call agents, softswitches, SIP proxies, H.323 gatekeepers, IP PBXs, etc., that are statically defined by IPv4 addresses. These network devices might also be IPv4 endpoints: SIP phones, IADs, MAs, media gateways, etc., that are defined by an IPv4 address prefix.

Realms support bandwidth-based call admission control and QoS marking for media. They are the basis for defining egress and ingress traffic to the Oracle Enterprise Session Border Controller—which supports the Oracle Enterprise Session Border Controller's topology hiding capabilities.

This chapter also explains how to configure media ports (steering pools). A steering pool exists within a realm and contains a range of ports that have a common address (for example, a target IPv4 address). The range of ports contained in the steering pool are used to steer media flows from one realm, through the Oracle Enterprise Session Border Controller, to another.

Finally, in this chapter you can learn about TOS/DiffServ functionality for realm-based packet marking by media type.

Overview

Realms are a logical distinction representing routes (or groups of routes) reachable by the Oracle Enterprise Session Border Controller and what kinds of resources and special functions apply to those routes. Realms are used as a basis for determining ingress and egress associations to network interfaces, which can reside in different VPNs. The ingress realm is determined by the signaling interface on which traffic arrives. The egress realm is determined by the following:

- Routing policy—Where the egress realm is determined in the session agent configuration or external address of a SIP-NAT
- Realm-bridging—As applied in the SIP-NAT configuration and H.323 stack configurations
- Third-party routing/redirect (i.e., SIP redirect or H.323 LCF)

Realms also provide configuration support for denial of service (DoS)/access control list (ACL) functionality.

Realms can also be nested in order to form nested realm groups. Nested realms consist of separate realms that are arranged within a hierarchy to support network architectures that have separate backbone networks and VPNs for signaling and media. This chapter provides detailed information about nested realms after showing you how to configure realms on your Oracle Enterprise Session Border Controller.

About Realms and Network Interfaces

All realms reference network interfaces on the Oracle Enterprise Session Border Controller. This reference is made when you configure a list of network interfaces in the realm configuration.

You configure a network interface to specify logical network interfaces that correspond existing physical interfaces on the Oracle Enterprise Session Border Controller. Configuring multiple network interfaces on a single physical interface creates a channelized physical interface, a VLAN. VLANs, in turn, allow you to reuse address space, segment traffic, and maximize bandwidth.

In order to reach the realms you configure, you need to assign them network interfaces. The values you set for the name and port in the network interface you select then indicate where the realm can be reached.

About the SIP Home Realm

The realm configuration is also used to establish what is referred to as the SIP home realm. This is the realm where the Oracle Enterprise Session Border Controller's SIP proxy sits.

In peering configurations, the SIP home realm is the internal network of the SIP proxy. In backbone access configurations, the SIP home realm typically interfaces with the backbone connected network. In additions, the SIP home realm is usually exposed to the Internet in an HNT configuration.

Although you configure a SIP home realm in the realm configuration, it is specified as the home realm in the main SIP configuration by the home realm identifier parameter. Specifying the SIP home realm means that the Oracle Enterprise Session Border Controller's SIP proxy can be addressed directly by connected entities, but other connected network signaling receives layer 3 NAT treatment before reaching the internal SIP proxy.

About Realms and Other Oracle Enterprise Session Border Controller Functions

Realms are referenced by other configurations in order to support this functionality across the protocols the Oracle Enterprise Session Border Controller supports and to make routing decisions. Other configurations' parameters that point to realms are:

- SIP configuration: home realm identifier, egress realm identifier
- SIP-NAT configuration: realm identifier
- H.323 stack configuration: realm identifier
- MGCP configuration: private realm, public realm
- Session agent configuration: realm identifier
- Media manager: home realm identifier
- Steering ports: realm identifier
- Static flow: in realm identifier, out realm identifier

Realms

Realm configuration is divided into the following functional areas, and the steps for configuring each are set out in this chapter: identity and IP address prefix, realm interfaces, realm service profiles, QoS measurement, QoS marking, address translation profiles, and DNS server configuration.

Before You Configure

Before you configure realms, you want to establish the physical and network interfaces with which the realm will be associated.

- Configure a physical interface to define the physical characteristics of the signaling line.
- Configure a network interface to define the network in which this realm is participating and optionally to create VLANs.

If you wish to use QoS, you should also determine if your Oracle Enterprise Session Border Controller is QoS enabled.

Remember that you will also use this realm in other configurations to accomplish the following:

- Set a signaling port or ports at which the Oracle Enterprise Session Border Controller listens for signaling messages.
- Configure sessions agents to point to ingress and egress signaling devices located in this realm in order to apply constraint for admission control.
- Configure session agents for defining trusted sources for accepting signaling messages.

Realm Configuration

To access the realm configuration parameters in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

   From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

Identity and IP Address Prefix

The first parameters you configure for a realm are its name (a unique identifier) and an IP address prefix and subnet mask.

The IP address and subnet mask establish a set of matching criteria for the realm, and distinguishes between realms that you assign to the same network interface.

To configure a realm's identity and IP address prefix in the ACLI:

1. identifier—Enter the name of the realm. This parameter uniquely identifies the realm. You will use this parameter in other configurations when asked for a realm identifier value.
2. addr-prefix—Enter the IPv4 or IPv6 address and subnet mask combination to set the criteria the Oracle Enterprise Session Border Controller uses to match packets sent or received on the network interface associated with this realm. This matching determines the realm, and subsequently what resources are used for that traffic. This setting determines whether the realm is an IPv4 or IPv6 realm.

   This parameter must be entered in the correct format where the IPv4 or IPv6 address comes first and is separated by a slash (/) from the subnet mask value. For example, 172.16.0.0/24.

   The default for this parameter is 0.0.0.0. When you leave this parameter set to the default, all addresses match.

Realm Interfaces

The realm points to one network interface on the Oracle Enterprise Session Border Controller.

Note: Only one network-interface can be assigned to a single realm-config object, except for Local multi-homing SCTP deployments.

To assign interfaces to a realm:

network-interfaces—Enter the physical and network interface(s) that you want this realm to reference. These are the network interfaces though which this realm can be reached by ingress traffic, and through which this traffic exits the system as egress traffic.

Enter the name and port in the correct format where the name of the interface comes first and is separated by a colon (:) from the port number. For example, f10:0.

The parameters you set for the network interfaces must be unique.

Enter multiple network interfaces for this list by typing an open parenthesis, entering each field value separated by a Space, typing a closed parenthesis, and then pressing Enter.

```
ACMEPACKET(realm-config)# network-interfaces fe1:0
```

You must explicitly configure a realm's network interface as either IPv4 or IPv6 when the applicable interface is either dual-stack or IPv6. You do this by appending the realm's network-interface with a .4 or a .6, as shown below.

```
ACMEPACKET(realm-config)# network-interfaces fe1:0.6
```

For single-stack interface configurations that do not specify this format, the Oracle Enterprise Session Border Controller assumes an IPv4 interface. Dual stack interface configurations fail if this IP version family suffix is not specified.

Realm Service Profile

The parameters you configure to establish the realm service profile determine how bandwidth resources are used and how media is treated in relation to the realm. Bandwidth constraints set for realm service profiles support the Oracle Enterprise Session Border Controller's admission control feature.

Peer-to-peer media between endpoints can be treated in one of three different ways:

- Media can be directed between sources and destinations within this realm on this specific Oracle Enterprise Session Border Controller. Media travels through the Oracle Enterprise Session Border Controller rather than straight between the endpoints.
- Media can be directed through the Oracle Enterprise Session Border Controller between endpoints that are in different realms, but share the same subnet.
- For SIP only, media can be released between multiple Oracle Enterprise Session Border Controllers.

  To enable SIP distributed media release, you must set the appropriate parameter in the realm configuration. You must also set the SIP options parameter to media-release with the appropriate header name and header parameter information. This option defines how the Oracle Enterprise Session Border Controller encodes IPv4 address and port information for media streams described by, for example, SDP.

  To configure realm service profile:

1. max-bandwidth—Enter the total bandwidth budget in kilobits per second for all flows to/from the realm defined in this element. The default is 0 which allows for unlimited bandwidth. The valid range is:
   - Minimum—0
   - Maximum—4294967295
2. mm-in-realm—Enable this parameter to treat media within this realm on this Oracle Enterprise Session Border Controller. The default is disabled. Valid values are:
   - enabled | disabled
3. mm-in-network—Enable this parameter to treat media within realms that have the same subnet mask on this Oracle Enterprise Session Border Controller. The default is enabled. Valid values are:
   - enabled | disabled
4. msm-release—Enable or disable the inclusion of multi-system (multiple Oracle Enterprise Session Border Controllers) media release information in the SIP signaling request sent into the realm identified by this realm-config element. If this field is set to enabled, another Oracle Enterprise Session Border Controller is allowed to decode the encoded SIP signaling request message data sent from a SIP endpoint to another SIP endpoint in the same network to resore the original SDP and subsequently allow the media to flow directly between those two SIP endpoints in the same network serviced by multiple Oracle Enterprise Session Border Controllers. If this field is disabled, the media and signaling will pass through boht Oracle Enterprise Session Border Controllers. Remember

that for this feature to work, you must also set the options parameter in the SIP configuration accordingly. The default is disabled. Valid values are:

- enabled | disabled

QoS Measurement

This chapter provides detailed information about when to configure the qos-enable parameter. If you are not using QoS or a QoS-capable Oracle Enterprise Session Border Controller, then you can leave this parameter set to disabled (default).

QoS Marking

QoS marking allows you to apply a set of TOS/DiffServ mechanisms that enable you to provide better service for selected networks

You can configure a realm to perform realm-based packet marking by media type, either audio/voice or video.

The realm configuration references a set of media policies that you configure in the media policy configuration. Within these policies, you can establish TOS/DiffServ values that define an individual type (or class) of service, and then apply them on a per-realm basis. In the media profiles, you can also specify:

- One or more audio media types for SIP and/or H.323
- One or more video types for SIP and/or H.323
- Both audio and video media types for SIP and/or H.323

To establish what media policies to use per realm in the ACLI:

media-policy—Enter the name (unique identifier) of the media policy you want to apply in the realm. When the Oracle Enterprise Session Border Controller first sets up a SIP or H.323 media session, it identifies the egress realm of each flow and then determines the media-policy element to apply to the flow. This parameter must correspond to a valid name entry in a media policy element. If you leave this parameter empty, then QoS marking for media will not be performed for this realm.

Address Translation Profiles

If you are not using this feature, you can leave the in-translationid and out-translationid parameters blank.

DNS Servers

You can configure DNS functionality on a per-network-interface basis, or you can configure DNS servers to use per realm. Configuring DNS servers for your realms means that you can have multiple DNS servers in connected networks. In addition, this allows you to specify which DNS server to use for a given realm such that the DNS might actually be in a different realm with a different network interface.

This feature is available for SIP and MGCP only.

To configure realm-specific DNS in the ACLI:

dns-realm—Enter the name of the network interface that is configured for the DNS service you want to apply in this realm. If you do not configure this parameter, then the realm will use the DNS information configured in its associated network interface.

DoS ACL Configuration

If you are not using this functionality, you can leave the parameters at their default values: average-rate-limit, peak-rate-limit, maximum-burst-size, access-control-trust-level, invalid-signal-threshold, and maximum-signal-threshold.

Enabling RTP-RTCP UDP Checksum Generation

You can configure the Oracle Enterprise Session Border Controller to generate a UDP checksum for RTP/ RTCP packets on a per-realm basis. This feature is useful in cases where devices performing network address translation

(NAT) do not pass through packets with a zero checksum from the public Internet. These packets do not make it through the NAT, even if they have the correct to and from IP address and UDP port information. When you enable this feature, the Oracle Enterprise Session Border Controller calculates a checksum for these packets and thereby enables them to traverse a NAT successfully.

If you do not enable this feature, then the Oracle Enterprise Session Border Controller will not generate a checksum for RTP or RTCP packets if their originator did not include one. If a checksum is already present when the traffic arrives at the hardware, the system will relay it.

You enable this feature on the outbound realm.

Aggregate Session Constraints Per Realm

You can set session constraints for the Oracle Enterprise Session Border Controller's global SIP configuration, specified session agents, and specified SIP interfaces. This forces users who have a large group of remote agents to create a large number of session agents and SIP interfaces.

With this feature implemented, however, you can group remote agents into one or more realms on which to apply session constraints.

To enable sessions constraints on a per realm basis:

constraint-name—Enter the name of the constraint you want to use for this realm. You set up in the session-constraints confiuration.

UDP Checksum Generation Configuration

To enable UDP checksum generation for a realm:

generate-udp-checksum—Enable this parameter to generate a UDP checksum for this outbound realm. The default is disabled. Valid values are:

- enabled | disabled

Admission Control Configuration

You can set admission control based on bandwidth for each realm by setting the max-bandwidth parameter for the realm configuration. Details about admission control are covered in this guide's *Admission Control and QoS* chapter.

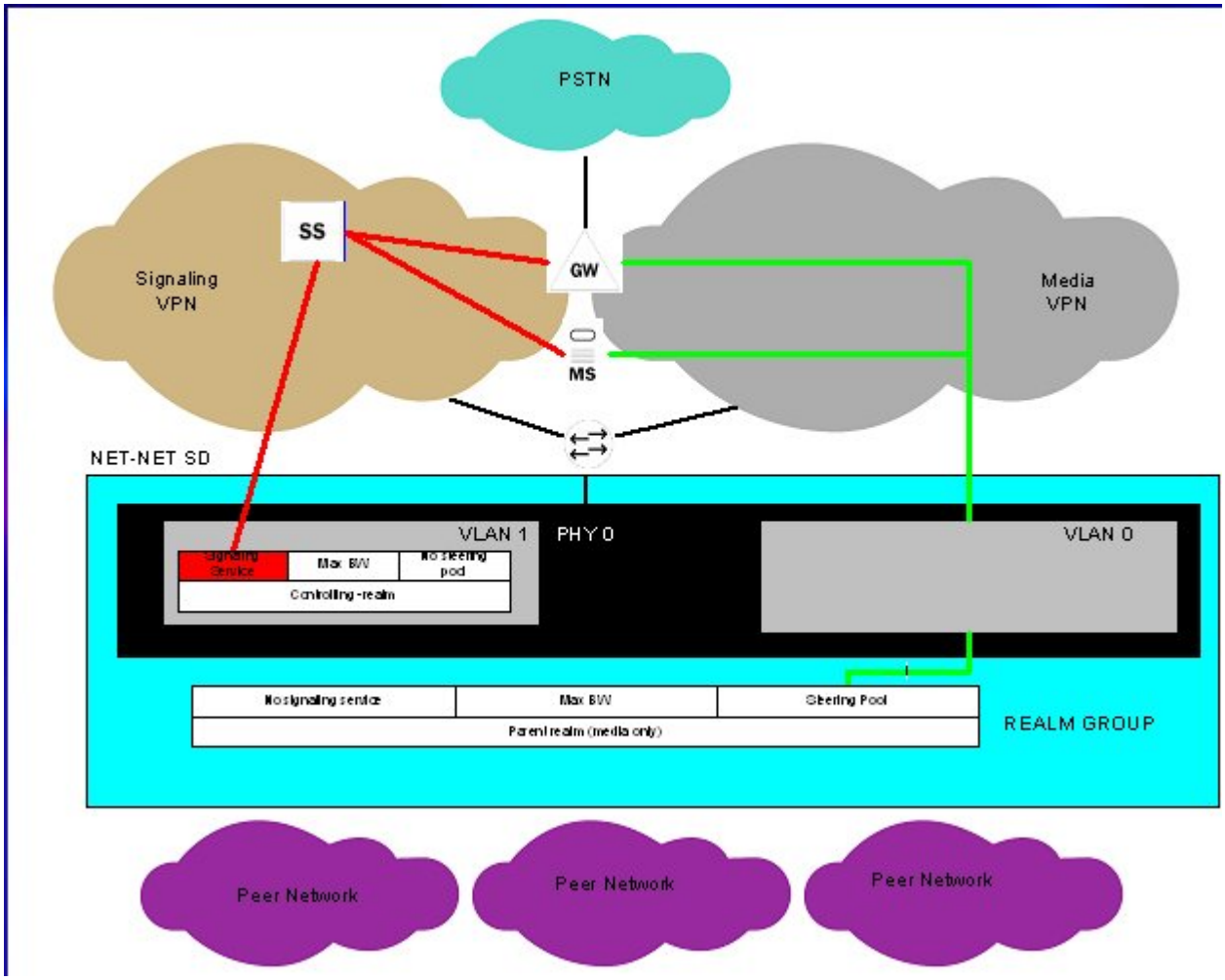
Reserved Parameters

In the ACLI, you do not need to configure the following parameters: max-latency, max-jitter, max-packet-loss, and observ-window-size.

Nested Realms

Configuring nested realms allows you to create backbone VPN separation for signaling and media. This means that you can put signaling and media on separate network interfaces, that the signaling and media VPN can have different address spaces, and that the parent realm has one media-only sub-realm.

The following figure shows the network architecture.



In addition, you can achieve enhanced scalability by using a shared service interface. A single service address is shared across many customers/peers, customer specific policies for bandwidth use and access control are preserved, and you can achieve fine-grained policy control.

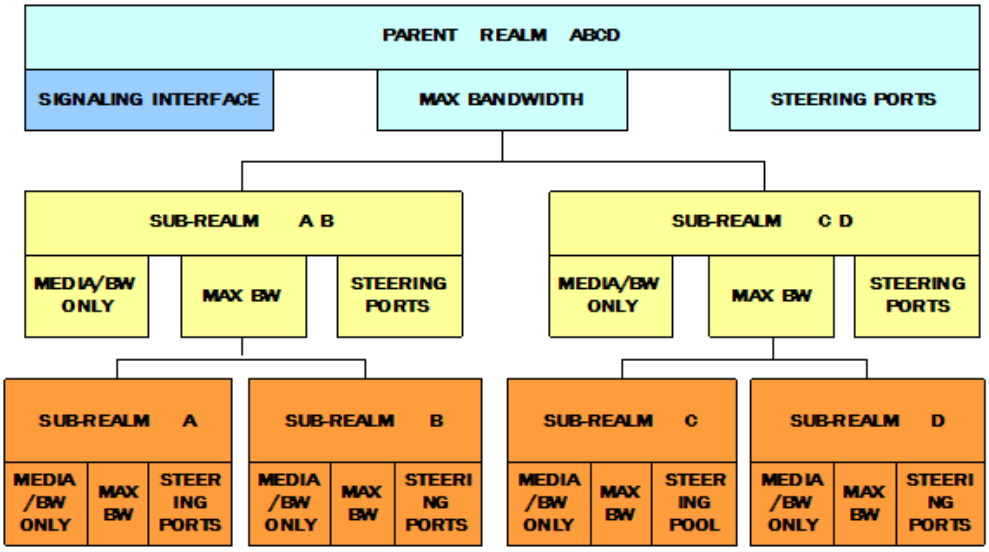
These benefits are achieved when you configure these types of realms:

- Realm group—A hierarchical nesting of realms identified by the name of the highest order realm.
- Controlling realm—A realms for which a signaling interface is configured. For example, you might configure these signaling interfaces in the following configurations: SIP-NAT, SIP port, H.323 stack, or MGCP. Typically, this is the highest order realm for the parent realm in a realm group.
- Parent realm—A realm that has one or more child realms. A parent realm might also be the child realm of another realm group.
- Child realm—A realm that is associated with a single higher order parent realm. A child might also be the parent realm of another realm group. Child realms inherit all signaling and steering ports from higher order realms.
- Media-only realm—A realm for which there is no configured signaling interface directly associated. Media-only realms are nested within higher order realms.

As these definitions suggest, parent and child realms can be constructed so that there are multiple nesting levels. Lower order realms inherit the traits of the realms above them, including: signaling service interfaces, session translation tables, and steering pools.

Since realms inherit the traits of the realms above them in the hierarchy, you will probably want to map what realms should be parents and children before you start configuring them. These relationships are constructed through one parameter in the realm configuration that identifies the parent realm for the configuration. If you specify a parent realm, then the realm you are configuring becomes a child realm subject to the configured parameters you have

established for that parent. And since parent realms can themselves be children of other realm, it is important that you construct these relationships with care.



Configuring Nested Realms

When you are configuring nested realms, you can separate signaling and media by setting realm parameters in the SIP interface configuration, the H.323 stack configuration, and the steering ports configuration.

- The realm identifier you set in the SIP interface configuration labels the associated realm for signaling.
- The realm identifier you set in the H.323 stack configuration labels the associated realm for signaling.
- The realm identifier you set in the steering ports configuration labels the associated realm for media.

For MGCP, you set a special option that enables nested realm use.

Constructing a hierarchy of nested realms requires that you note which realms you want to handle signaling, and which you want to handle media.

In the SIP port configuration for the SIP interface and in the H.323 stack configuration, you will find an allow anonymous parameter that allows you to set certain access control measures. The table below outlines what each parameter means.

Allow Anonymous Parameter	Description
all	All anonymous connections allowed.
agents-only	Connections only allowed from configured session agents.
realm-prefix	Connections only allowed from addresses with the realm's address prefix and configured session agents.
registered	Connections allowed only from session agents and registered endpoints. (For SIP only, a REGISTER is allowed for any endpoint.)
register-prefix	Connections allowed only from session agent and registered endpoints. (For SIP only, a REGISTER is allowed for session agents and a matching realm prefix.)

Parent and Child Realm Configuration

To configure nested realms, you need to set parameters in the realm configuration.

To configure parent and child realms:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. parent-realm—Enter the identifier of the realm you want to name as the parent. Configuring this parameter makes the realm you are currently configuring as the child of the parent you name. As such, the child realm is subject to the configured parameters for the parent.

Required Signaling Service Parameters

To configure nested realms, you need to set parameters in the realm configuration and in the configurations for the signaling protocols you want to use.

To configure H.323 stack parameters for nested realms:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. Type h323-stacks and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   From this point, you can configure H.323 stack parameters. To view all h323-stack configuration parameters, enter a ? at the system prompt.

5. allow-anonymous—Enter the admission control of anonymous connections accepted and processed by this H.323 stack. The default is all. The valid values are:

   - all—Allow all anonymous connections
   - agents-only—Only requests from session agents allowed
   - realm-prefix—Session agents and address matching relam prefix

Aggregate Session Constraints Nested Realms

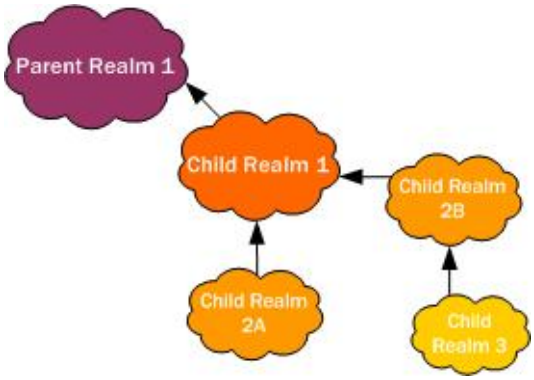
In addition to setting session constraints per realm for SIP and H.323 sessions, you can also enable the Oracle Enterprise Session Border Controller to apply session constraints across nested realms. When you set up session constraints for a realm, those constraints apply only to the realm for which they are configured without consideration for its relationship either as parent or child to any other realms.

You can also, however, enable the Oracle Enterprise Session Border Controller to take nested realms into consideration when applying constraints. For example, if a call enters on a realm that has no constraints but its parent does, then the constraints for the parent are applied. This parameter is global and so applies to all realms on the system. For the specific realm the call uses and for all of its parents, the Oracle Enterprise Session Border Controller increments the counters upon successful completion of an inbound or outbound call.

In the following example, you can see one parent realm and its multiple nested, child realms. Now consider applying these realm constraints:

- Parent Realm 1—55 active sessions
- Child Realm 1—45 active sessions
- Child Realm 2A—30 active sessions

- Child Realm 2B—90 active sessions
- Child Realm 3—20 active sessions



Given the realm constraints outlined above, consider these examples of how global session constraints for realms. For example, a call enters the Oracle Enterprise Session Border Controller on Child Realm 2B, which has an unmet 90-session constraint set. Therefore, the Oracle Enterprise Session Border Controller allows the call based on Child Realm 2B. But the call also has to be within the constraints set for Child Realm 1 and Parent Realm 1. If the call fails to fall within the constraints for either of these two realms, then the Oracle Enterprise Session Border Controller rejects the call.

Impact to Other Session Constraints and Emergency Calls

You can set up session constraints in different places in your Oracle Enterprise Session Border Controller configuration. Since session agents and SIP interfaces also take session constraints, it is import to remember the order in which the Oracle Enterprise Session Border Controller applies them:

1. Session agent session constraints
2. Realm session constraints (including parent realms)
3. SIP interface session constraints

   Emergency and priority calls for each of these is exempt from session constraints. That is, any call coming into the Oracle Enterprise Session Border Controller marked priority is processed.

Session Contraints Configuration

You enabled use of session constraints for nested realms across the entire system by setting the nested-realms-stats parameter in the session router configuration to enabled.

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type session-router and press Enter.

   ```
   ACMEPACKET(session-router)# session-router
   ACMEPACKET(session-router-config)#
   ```

4. nested-realms-stats—Change this parameter from disabled (default) to enabled if you want the Oracle Enterprise Session Border Controller to apply session constraints across all nested realms (realms that are children to other realms)
5. Save and activate your configuration.

Realm-Based Packet Marking

The Oracle Enterprise Session Border Controller supports TOS/DiffServ functions that allow you to

- Set up realm-based packet marking by media type, either audio-voice or video
- Set up realm-based packet marking for signaling, either SIP or H.323

Upstream devices use these markings to classify traffic in order to determine the priority level of treatment it will receive.

About TOS DiffServ

TOS and DiffServ are two different mechanisms used to achieve QoS in enterprise and service provider networks; they are two different ways of marking traffic to indicate its priority to upstream devices in the network.

For more information about TOS (packet) marking, refer to:

- IETF RFC 1349 (*http://www.ietf.org/rfc/rfc1349.txt*)

For more information about DiffServ, refer to:

- IETF RFC 2474 (*http://www.ietf.org/rfc/rfc2474.txt*)
- IETF RFC 2475 (*http://www.ietf.org/rfc/rfc2475.txt*).

ToS Byte

The TOS byte format is as follows:

The TOS byte is broken down into three components:

- Precedence—The most used component of the TOS byte, the precedence component is defined by three bits. There are eight possible precedence values ranging from 000 (decimal 0) through 111 (decimal 7). Generally, a precedence value of 000 refers to the lowest priority traffic, and a precedence value of 111 refers to the highest priority traffic.
- TOS—The TOS component is defined by four bits, although these bits are rarely used.
- MBZ—The must be zero (MBZ) component of the TOS byte is never used.

DiffServ Byte

Given that the TOS byte was rarely used, the IETF redefined it and in doing so created the DiffServ byte.

The DiffServ byte format is as follows:

The DiffServ codepoint value is six bits long, compared to the three-bit-long TOS byte's precedence component. Given the increased bit length, DiffServ codepoints can range from 000000 (decimal 0) to 111111 (decimal 63).

Note: By default, DiffServ codepoint mappings map exactly to the precedence component priorities of the original TOS byte specification.

Packet Marking for Media

You can set the TOS/DiffServ values that define an individual type or class of service for a given realm. In addition, you can specify:

- One or more audio media types for SIP and/or H.323
- One or more video media types for SIP and/or H.323

- Both audio and video media types for SIP and/or H.323

For all incoming SIP and H.23 requests, the media type is determined by negotiation or by preferred codec. SIP media types are determined by the SDP, and H.323 media types are determined by the media specification transmitted during call setup.

Configuring Packet Marking by Media Type

This section describes how to set up the media policy configuration that you need for this feature, and then how to apply it to a realm.

These are the ACLI parameters that you set for the media policy:

```
name         media policy name
tos-settings  list of TOS settings
```

This is the ACLI parameter that you set for the realm:

```
media-policy default media policy name
```

Packet Marking Configuration

To set up a media policy configuration to mark audio-voice or video packets:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type media-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# media-policy
   ACMEPACKET(media-policy)#
   ```

   From this point, you can configure media policy parameters. To view all configuration parameters for media profiles, enter a ? at the system prompt.

4. Type media-policy and press Enter.

   ```
   ACMEPACKET(media-manager)# media-policy
   ```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration you want to edit.

5. name—Create a reference name for this policy and press Enter.
6. Type tos-settings and press Enter.

   ```
   ACMEPACKET(media-policy)# tos-settings
   ```

7. media-type—Enter the media type that you want to use for this group of TOS settings. You can enter any of the IANA-defined media types for this value: audio, example, image, message, model, multipart, text, and video. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

   ```
   ACMEPACKET(tos-settings)# media-type message
   ```

8. media-sub-type—Enter the media sub-type you want to use for the media type. This value can be any of the sub-types that IANA defines for a specific media type. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

   ```
   ACMEPACKET(tos-settings)# media-sub-type sip
   ```

9. media-attributes—Enter the media attribute that will match in the SDP. This parameter is a list, so you can enter more than one value. The values are case-sensitive and can be up to 255 characters in length. This parameter has no default.

   If you enter more than one media attribute value in the list, then you must enclose your entry in quotation marks ().

```
ACMEPACKET(tos-settings)# media-attributes sendonly sendrecv
```

10. tos-value—Enter the TOS value you want applied for matching traffic. This value is a decimal or hexidecimal value. The valid range is:

    - 0x00 to 0xFF.

    ```
    ACMEPACKET(tos-settings)# tos-value 0xF0
    ```

11. Save and activate your configuration.

Applying a Media Policy to a Realm

To apply a media policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm
   ACMEPACKET(realm)#
   ```

4. media-policy—Enter the unique name of the media policy you want to apply to this realm.

Signaling Packet Marking Configuration

ToS marking for signaling requires you to configure a media policy and set the name of the media policy in the appropriate realm configuration.

This section shows you how to configure packet marking for signaling.

Configuring a Media Policy for Signaling Packet Marking

To set up a media policy configuration to mark signaling packets:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type media-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# media-policy
   ACMEPACKET(media-policy)#
   ```

   From this point, you can configure media policy parameters. To view all media policy configuration parameters, enter a ? at the system prompt.

4. Type media-policy and press Enter.

   ```
   ACMEPACKET(media-manager)# media-policy
   ```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration you want to edit.

5. name—Create a reference name for this policy and press Enter.
6. Type tos-settings and press Enter.

   ```
   ACMEPACKET(media-policy)# tos-settings
   ```

7. media-type—Enter the media type that you want to use for this group of TOS settings. You can enter any of the IANA-defined media types for this value: audio, example, image, message, model, multipart, text, and video. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

```
ACMEPACKET(tos-settings)# media-type message
```

8. media-sub-type—Enter the media sub-type you want to use for the media type. This value can be any of the sub-types that IANA defines for a specific media type. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

```
ACMEPACKET(tos-settings)# media-sub-type sip
```

9. media-attributes—Enter the media attribute that will match in the SDP. This parameter is a list, so you can enter more than one value. The values are case-sensitive and can be up to 255 characters in length. This parameter has no default.

   If you enter more than one media attribute value in the list, then you must enclose your entry in quotation marks ().

```
ACMEPACKET(tos-settings)# media-attributes sendonly sendrecv
```

10. tos-value—Enter the TOS value you want applied for matching traffic. This value is a decimal or hexidecimal value. The valid range is:

    - 0x00 to 0xFF.

```
ACMEPACKET(tos-settings)# tos-value 0xF0
```

11. Save and activate your configuration.

Applying a Media Policy to a Realm

To apply a media policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```

4. media-policy—Enter the unique name of the media policy you want to apply to this realm.

Using Class Profile for Packet Marking

Class profile provides an additional means of ToS marking, but only for limited circumstances. Use class-profile only if you are marking ToS on traffic destined for a specific To address, and when media-policy is not used on the same realm. Using media-policy for ToS marking is, by far, more common.

To configure a class profile, you prepare your desired media policy, create the class profile referencing the media policy and the To address, and set the name of the class profile in the appropriate realm configuration.

Class Profile and Class Policy Configuration

This section shows you how to configure packet marking using a class profile.

To configure the class profile and class policy:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type class-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# class-profile
ACMEPACKET(class-profile)#
```

4. Type policy and press Enter to begin configuring the class policy.

```
ACMEPACKET(class-profile)# policy
```

From this point, you can configure class policy parameters. To view all class policy configuration parameters, enter a ? at the system prompt.

5. profile-name—Enter the unique name of the class policy. When you apply a class profile to a realm configuration, you use this value.
6. to-address—Enter a list of addresses to match to incoming traffic for marking. You can use E.164 addresses, a host domain address, or use an asterisk (*) to set all host domain addresses.
7. media-policy—Enter the name of the media policy you want to apply to this class policy.

Applying a Class Policy to a Realm

To apply a class policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type media-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```

4. class-profile—Enter the name if the class profile to apply to this realm. This is the name you set in the profile-name parameter of the class-policy configuration.

SIP-SDP DCSP Marking ToS Bit Manipulation

Used to indicate priority and type of requested service to devices in the network, type of service (TOS) information is included as a set of four-bit flags in the IP header. Each bit has a different purpose, and only one bit at a time can be set: There can be no combinations. Available network services are:

- Minimum delay—Used when latency is most important
- Maximum throughput—Used when the volume of transmitted data in any period of time is important
- Maximum reliability—Used when it is important to assure that data arrives at its destination without requiring retransmission
- Minimum cost—Used when it is most important to minimize data transmission costs

The Oracle Enterprise Session Border Controller's support for type of service (TOS allows you to base classification on the media type as well as the media subtype. In prior releases, you can configure the Oracle Enterprise Session Border Controller to mark TOS bits on outgoing packets using a media policy. Supported media types include audio, video, application, data, image, text, and message; supported protocol types are H.225, H.245, and SIP. Note that, although H.225 and H.245 are not part of any IANA types, they are special cases (special subtypes) of message for the Oracle Enterprise Session Border Controller. When these criteria are met for an outgoing packet, the Oracle Enterprise Session Border Controller applies the TOS settings to the IP header. The augmented application of TOS takes matching on media type or protocol and expands it to match on media type, media-sub-type, and media attributes.

The new flexibility of this feature resolves issues when, for example, a customer needs to differentiate between TV-phone and video streaming. While both TV-phone and video streaming have the attribute "media=video," TV-phone streaming has "direction=sendrcv" prioritized at a high level and video has direction=sendonly or recvonly with middle level priority. The Oracle Enterprise Session Border Controller can provide the appropriate marking required to differentiate the types of traffic.

In the media policy, the tos-values parameter accepts values that allow you to create any media type combination allowed by IANA standards. This is a dynamic process because theOracle Enterprise Session Border Controller generates matching criteria directly from messages.

The new configuration takes a media type value of any of these: audio, example, image, message, model, multipart, text, and video. It also takes a media sub-type of any value specified for the media type by IANA; however, support for T.38 must be entered exactly as t.38 (rather than t38). Using these values, the Oracle Enterprise Session Border Controller creates a value Based on a combination of these values, the Oracle Enterprise Session Border Controller applies TOS settings.

You also configure the TOS value to be applied, and the media attributes you want to match.

You can have multiple groups of TOS settings for a media policy.

ToS Bit Manipulation Configuration

This section provides instructions for how to configure TOS bit manipulation on your Oracle Enterprise Session Border Controller.

To configure TOS bit manipulation:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type media-policy and press Enter.

   ```
   ACMEPACKET(media-manager)# media-policy
   ```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration you want to edit.

4. name—Create a reference name for this policy and press Enter.
5. Type tos-settings and press Enter.

   ```
   ACMEPACKET(media-policy)# tos-settings
   ```

6. media-type—Enter the media type that you want to use for this group of TOS settings. You can enter any of the IANA-defined media types for this value: audio, example, image, message, model, multipart, text, and video. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

   ```
   ACMEPACKET(tos-settings)# media-type message
   ```

7. media-sub-type—Enter the media sub-type you want to use for the media type. This value can be any of the sub-types that IANA defines for a specific media type. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

   ```
   ACMEPACKET(tos-settings)# media-sub-type sip
   ```

8. media-attributes—Enter the media attribute that will match in the SDP. This parameter is a list, so you can enter more than one value. The values are case-sensitive and can be up to 255 characters in length. This parameter has no default.

   If you enter more than one media attribute value in the list, then you must enclose your entry in quotation marks ().

   ```
   ACMEPACKET(tos-settings)# media-attributes sendonly sendrecv
   ```

9. tos-value—Enter the TOS value you want applied for matching traffic. This value is a decimal or hexidecimal value. The valid range is:

   - 0x00 to 0xFF.

     ```
     ACMEPACKET(tos-settings)# tos-value 0xF0
     ```

10. Save and activate your configuration.

Steering Pools

Steering pools define sets of ports that are used for steering media flows through the Oracle Enterprise Session Border Controller. These selected ports are used to modify the SDP to cause receiving session agents to direct their media toward this system. Media can be sent along the best quality path using these addresses and ports instead of traversing the shortest path or the BGP-4 path.

For example, when the Oracle Enterprise Session Border Controller is communicating with a SIP device in a specific realm defined by a steering pool, it uses the IP address and port number from the steering pool's range of ports to direct the media. The port the Oracle Enterprise Session Border Controller chooses to use is identified in the SDP part of the message.

Note: The values entered in the steering pool are used when the system provides NAT, PAT, and VLAN translation.

Configuration Overview

To plan steering pool ranges, take into account the total sessions available on the box, determine how many ports these sessions will use per media stream, and assign that number of ports to all of the steering pools on your Oracle Enterprise Session Border Controller. For example, if your Oracle Enterprise Session Border Controller can accommodate 500 sessions and each session typically uses 2 ports, you would assign 1000 ports to each steering pool. This strategy provides for a maximum number of ports for potential use, without using extra resources on ports your Oracle Enterprise Session Border Controller will never use.

The following table lists the steering pool parameters you need to configure:

Parameter	Description
IP address	IPv4 address of the steering pool.
start port	Port number that begins the range of ports available to the steering pool. You must define this port to enable the system to perform media steering and NAT operations.
end port	Port number that ends the range of ports available to the steering pool. You must define this port to enable the system to perform media steering and NAT operations.
realm id	Identifies the steering pool's realm. The steering pool is restricted to only the flows that originate from this realm.

Note: The combination of entries for IP address, start port, and realm ID must be unique in each steering pool. You cannot use the same values for multiple steering pools.

Each bidirectional media stream in a session uses two steering ports, one in each realm (with the exception of audio/video calls that consume four ports). You can configure the start and end port values to provide admission control. If all of the ports in all of the steering pools defined for a given realm are in use, no additional flows/sessions can be established to/from the realm of the steering pool.

Steering Pool Configuration

To configure a steering pool:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type steering-pool and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# steering-pool
ACMEPACKET(steering-pool)#
```

4. ip-address—Enter the target IPv4 address of the steering pool in IP address format. For example:

```
192.168.0.11
```

5. start-port—Enter the start port value that begins the range of ports available to this steering pool. The default is 0. The valid range is:

   - Minimum—0
   - Maximum—65535

   You must enter a valid port number or the steering pool will not function properly.

6. end-port—Enter the end port value that ends the range of ports available to this steering pool. The default is 0. The valid range is:

   - Minimum—0
   - Maximum—65535

   You must enter a valid port number or the steering pool will not function properly.

7. realm-id—Enter the realm ID to identify the steering pool's realm, following the name format. The value you enter here must correspond to the value you entered as the identifier (name of the realm) when you configured the realm. For example:

```
peer-1
```

   This steering pool is restricted to flows that originate from this realm.

   The following example shows the configuration of a steering pool that

```
steering-pool
        ip-address                    192.168.0.11
        start-port                    20000
        end-port                      21000
        realm-id                      peer-1
        last-modified-date            2005-03-04 00:35:22
```

SDP Alternate Connectivity

The Oracle Enterprise Session Border Controller can create an egress-side SDP offer containing both IPv4 and IPv6 media addresses via a mechanism which allows multiple IP addresses, of different address families (i.e., IPv4 & IPv6) in the same SDP offer. Our implementation is based on the RFC draft "draft-boucadair-mmusic-altc-09".

Each realm on the Oracle Enterprise Session Border Controller can be configured with an alternate family realm on which to receive media in the alt family realm parameter in the realm config. As deployed, one realm will be IPv4, and the alternate will be IPv6. The Oracle Enterprise Session Border Controller creates the outbound INVITE with IPv4 and IPv6 addresses to accept the media, each in an a=altc: line and each in its own realm. The IP addresses inserted into the a=altc: line are from the egress realm's and alt-realm-family realm's steering pools. Observe in the image how the red lines indicate the complementary, alternate realms.

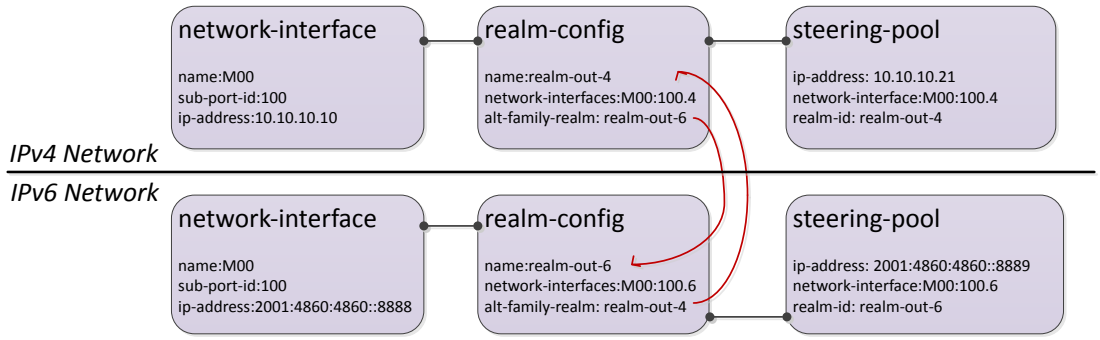
You can configure the order in which the a=altc: lines appear in the SDP in the pref-address-type parameter in the realm-config. This parameter can be set to

- IPv4 - SDP contains the IPv4 address first
- IPv6 - SDP contains the IPv6 address first
- NONE - SDP contains the native address family of the egress realm first

In the 200OK to the INVITE, the callee chooses either the IPv6 or IPv4 address to use for the call's media transport between itself and Oracle Enterprise Session Border Controller. After the Oracle Enterprise Session Border Controller receives the 200OK, the chosen flow is installed, and the unused socket is discarded.

For two realms from different address families to share the same physical interface and vlan, you use a .4 or .6 tag in the network-interface reference. When IPv4 and IPv6 realms share the same network-interface and VLAN, you identify them by realm name and network-interface configured as:

- IPv4 - <phy-interface>:<vlan>.4
- IPv6 - <phy-interface>:<vlan>.6



If the INVITE's egress realm is IPv6, pref-address-type = NONE, the outbound SDP has these a=altc: lines:

```
a=altc:1 IPv6 2001:4860:4860::8889 20001
a=altc:2 IPv4 10.10.10.21 20001
```

If the INVITE's egress realm is IPv6, pref-address-type = IPv4, the outbound SDP has these a=altc: lines:

```
a=altc:1 IPv4 10.10.10.21 20001
a=altc:2 IPv6 2001:4860:4860::8889 20001
```

SDP Alternate connectivity supports B2B and hairpin call scenarios. SDP Alternate connectivity also supports singleterm, B2B, and hairpin call scenarios.

When providing SDP alternate connectivity for SRTP traffic, in the security policy configuration element, the network-interface parameter's value must be configured with a .4 or .6 suffix to indicate IPv4 or IPv6 network, respectively.

SDP Alternate Connectivity Configuration

To configure SDP alternate connectivity:

1. Access the **realm-config** configuration element.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

2. Select the **realm-config** object to edit.

   ```
   ACMEPACKET(realm-config)# select
   identifier:
   1: realm01 left-left:0 0.0.0.0

   selection: 1
   ACMEPACKET(realm-config)#
   ```

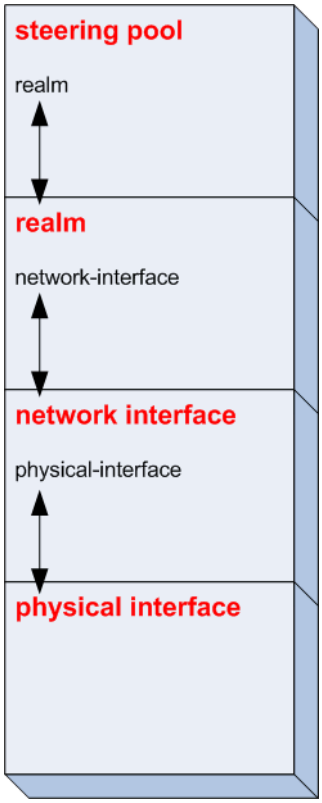
3. alt-realm-family — Enter the realm name of the alternate realm, from which to use an IP address in the other address family. If this parameter is within an IPv4 realm configuration, you will enter an IPv6 realm name.
4. pref-address-type — Set the order in which the a=altc: lines suggest preference. Valid values are:
   - none — address family type of egress realm signaling
   - ipv4 — IPv4 realm/address first
   - ipv6 — IPv6 realm/address first
5. Type **done** to save your configuration.

Multiple Interface Realms

The multi-interface realm feature lets you group multiple network interfaces to aggregate their bandwidth for media flows. In effect, this feature lets you use the total throughput of the available physical interfaces on your Oracle Enterprise Session Border Controller for a single realm. Multi-interface realms are implemented by creating multiple steering pools, each on an individual network interface, that all reference a single realm.

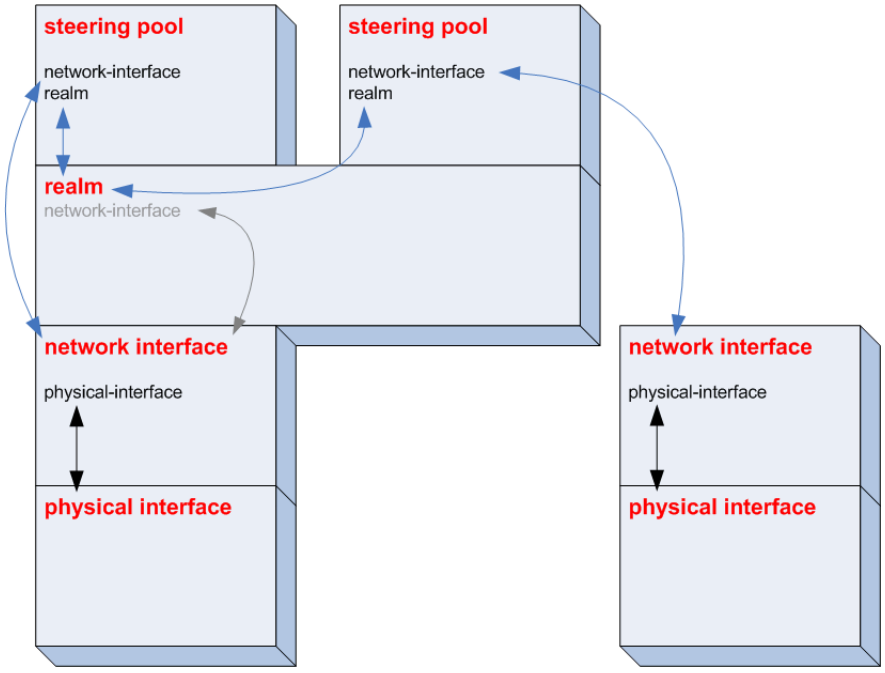
Of course, you can not to use this feature and configure your Oracle Enterprise Session Border Controller to create a standard one-realm to one-network interface configuration.

Without using multiple interface realms, the basic hierarchical configuration of the Oracle Enterprise Session Border Controller from the physical interface through the media steering pool looks like this:

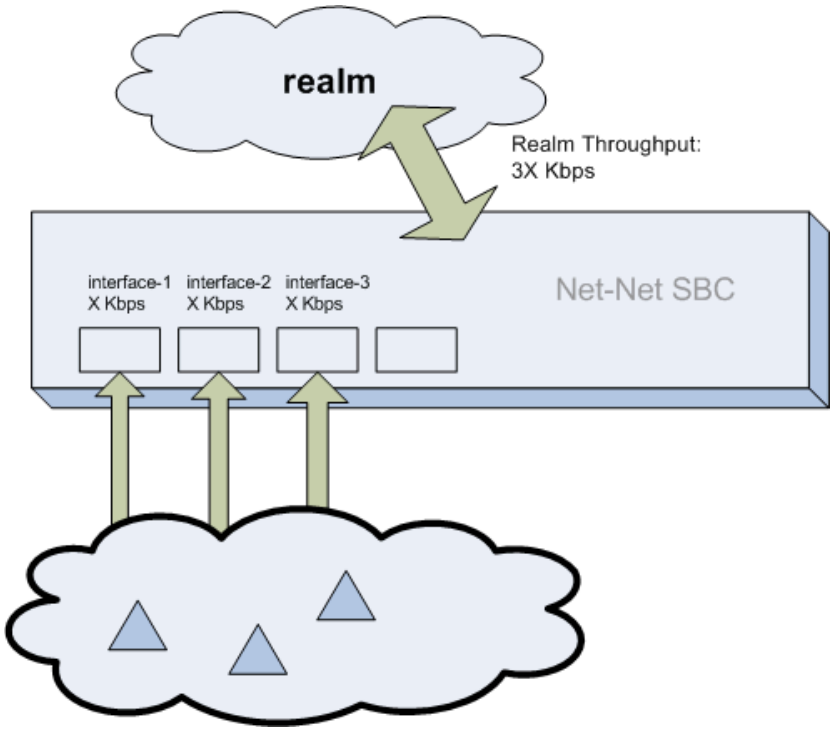


In this model, one (non-channelized) network interface exists on a physical interface. One realm exists on one network interface. One or more steering pools can exist on one realm. Within each higher level configuration element exists a parameter that references a lower level configuration element in the Oracle Enterprise Session Border Controller's logical network model.

The multi-interface realm feature directs media traffic entering and exiting multiple network interfaces in and out of a single realm. Since all the steering pools belong to the same realm, their assigned network interfaces all feed into the same realm as well. The following diagram shows the relationship in the new logical model:



The advantage of using multi-interface realms is the ability to aggregate the bandwidth available to multiple network interfaces for a larger-than-previously-available total bandwidth for a realm. In the illustration below, three physical interfaces each have X Kbps of bandwidth. The total bandwidth available to the realm with multiple network interfaces is now 3X the bandwidth. (In practical usage, interface-1 only contributes X - VoIP Signaling to the total media bandwidth available into the realm.)



Steering Pool Port Allocation

Every steering pool you create includes its own range of ports for media flows. The total number of ports in all the steering pools that feed into one realm are available for calls in and out of the realm.

Steering pool ports for a given realm are assigned to media flows sequentially. When the first call enters the Oracle Enterprise Session Border Controller after start-up, it is assigned the first ports on the first steering pool that you configured. New calls are assigned to ports sequentially in the first steering pool. When all ports from the first steering pool are exhausted, the Oracle Enterprise Session Border Controller uses ports from the next configured steering pool. This continues until the last port on the last configured steering pool is used.

After the final port is used for the first time, the next port chosen is the one first returned as empty from the full list of ports in all the steering pools. As media flows are terminated, the ports they used are returned to the realm's full steering pool. In this way, after initially exhausting all ports, the realm takes new, returned, ports from the pool in a least last used manner.

When a call enters the Oracle Enterprise Session Border Controller, the signaling application allocates a port from all of the eligible steering pools that will be used for the call. Once a port is chosen, the Oracle Enterprise Session Border Controller checks if the steering pool that the port is from has a defined network interface. If it does, the call is set up on the corresponding network interface. If a network interface is not defined for that steering pool, the network interface defined for the realm is used.

Network Interface Configuration

This section explains how to configure your Oracle Enterprise Session Border Controller to use multiple interface realms.

You must first configure multiple physical interfaces and multiple network interfaces on your Oracle Enterprise Session Border Controller.

To configure the realm configuration for multi-interface realms.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter. The system prompt changes.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

   From this point, you can configure a realm that will span multiple network interfaces.

4. network-interfaces—Enter the name of the network interface where the signaling traffic for this realm will be received.

Creating Steering Pools for Multiple Interface Realms

To configure steering pools for multi-interface realms:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type steering-pool and press Enter. The system prompt changes.

```
ACMEPACKET(media-manager)# steering-pool
ACMEPACKET(steering-pool)#
```

   From this point, you can configure steering pools which collectively bridge the multiple network interfaces they are connected to.

4. ip-address—Enter the IP address of the first steering pool on the first network interface.

   This IP address must correspond to an IP address within the subnet of a network interface you have already configured.

This IP can not exist on a network interface other than the one you configure in the network-interface parameter.

5. start-port—Enter the beginning port number of the port range for this steering pool. The default is 0. The valid range is:
   - Minimum—0
   - Maximum—65535
6. end-port—Enter the ending port number of the port range for this steering pool. The default is 0. The valid range is:
   - Minimum—0
   - Maximum—65535
7. realm-id—Enter the name of the realm which this steering pool directs its media traffic toward.
8. network-interface—Enter the name of the network interface you want this steering pool to direct its media toward. This parameter will match a name parameter in the network-interface configuration element. If you do not configure this parameter, you can only assign a realm to a single network interface, as the behavior was in all Oracle Enterprise Session Border Controller Software releases pre- 2.1.
9. Create additional steering pools on this and on other network interfaces as needed. Remember to type done when you are finished configuring each new steering pool.

Media over TCP

The Oracle Enterprise Session Border Controller now supports RFC 4145 (TCP-Based Media Transport in the SDP), also called TCP Bearer support. Media over TCP can be used to support applications that use TCP for bearer path transport.

RFC 4145 adds two new attributes, setup and connection, to SDP messages. The setup attribute indicates which end of the TCP connection should initiate the connection. The connection attribute indicates whether an existing TCP connection should be used or if a new TCP connection should be setup during re-negotiation. RFC 4145 follows the offer/answer model specified in RFC3264. An example of the SDP offer message from the end point 192.0.2.2 as per RFC4145 is as given below:

```
m=image 54111 TCP t38
c=IN IP4 192.0.2.2
a=setup:passive
a=connection:new
```

This offer message indicates the availability of t38 fax session at port 54111 which runs over TCP. Oracle Enterprise Session Border Controller does not take an active part in the application-layer communication between each endpoint.

The Oracle Enterprise Session Border Controller provides the means to set up the end-to-end TCP flow by creating the TCP/IP path based on the information learned in the SDP offer/answer process.

TCP Bearer Conditions

The following conditions are applicable to the Oracle Enterprise Session Border Controller's support of RFC 4145.

1. The Oracle Enterprise Session Border Controller can not provide media-over-TCP for HNT scenarios (endpoints behind a NAT).
2. When media is released into the network, the TCP packets do not traverse the Oracle Enterprise Session Border Controller because nNo TCP bearer connection is created.
3. The Oracle Enterprise Session Border Controller does not inspect the setup and connection attributes in the SDP message since the TCP packets transparently pass through the Oracle Enterprise Session Border Controller. These SDP attributes are forwarded to the other endpoint. It is the other endpoint's responsibility to act accordingly.
4. After the Oracle Enterprise Session Border Controllerreceives a SYN packet, it acts as a pure pass through for that TCP connection and ignores all further TCP handshake messages including FIN and RST. The flow will only be torn down in the following instances:

- The TCP initial guard timer, TCP subsequent guard timer, or the TCP flow time limit timer expire for that flow.
- The whole SIP session is torn down.

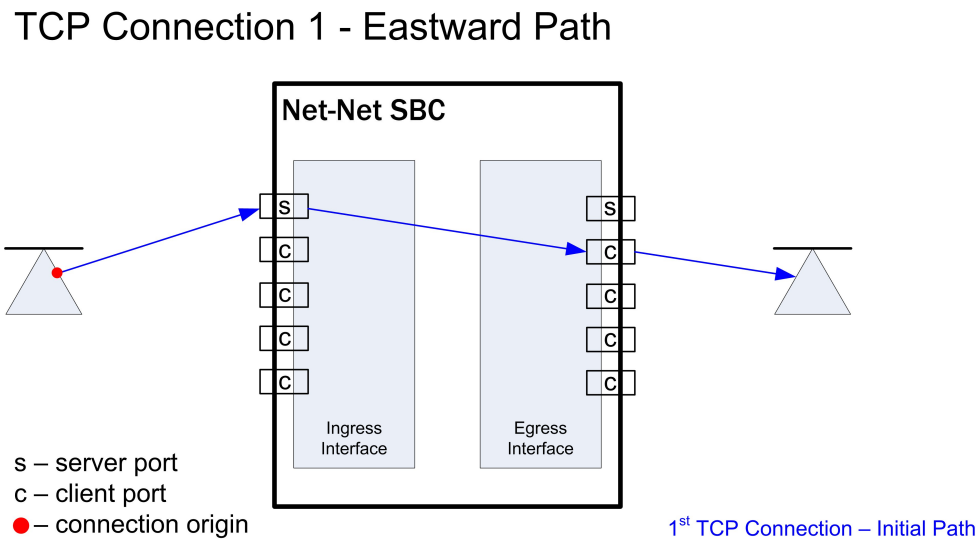
TCP Port Selection

When a call is first set up, the Oracle Enterprise Session Border Controller inspects the SDP message's m-line to see if any media will be transported via TCP. If the SDP message indicates that some content will use TCP, the Oracle Enterprise Session Border Controller allocates a configured number of steering ports for the media-over-TCP traffic. These TCP media ports are taken from the each realm's steering pool.

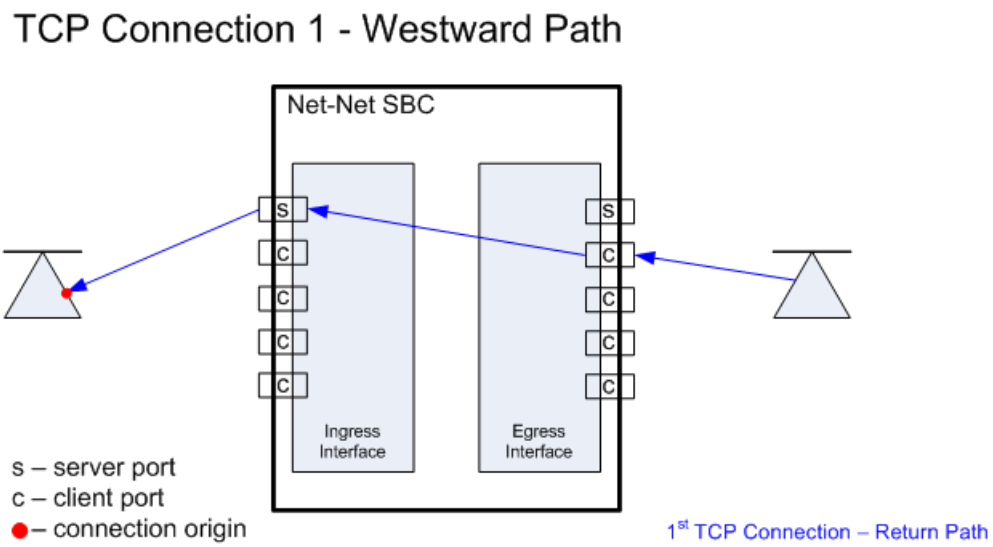
Each endpoint can initiate up to four end-to-end TCP flows between itself and the other endpoint. The Oracle Enterprise Session Border Controller assigns one port to receive the initial TCP packet (server port), and one to four ports assigned to send TCP traffic (client ports) to the receiving side of the TCP flow. The number of TCP flows for each call is configured globally.

In order to configure the Oracle Enterprise Session Border Controller to facilitate and support this process, you need to specify the number of ports per side of the call that can transport discrete TCP flows. You can configure one to four ports/flows. For configuration purposes, the Oracle Enterprise Session Border Controller counts this number as inclusive of the server port. Therefore if you want the Oracle Enterprise Session Border Controller to provide a maximum of one end-to-end TCP flow, you have to configure two TCP ports; one to receive, and one to send. The receiving port (server) is reused to set up every flow, but the sending port (client) is discrete per flow. For example: for 2 flows in each direction, set the configuration to 3 TCP ports per flow; for 3 flows in each direction, set the configuration to 4 TCP ports per flow, etc.

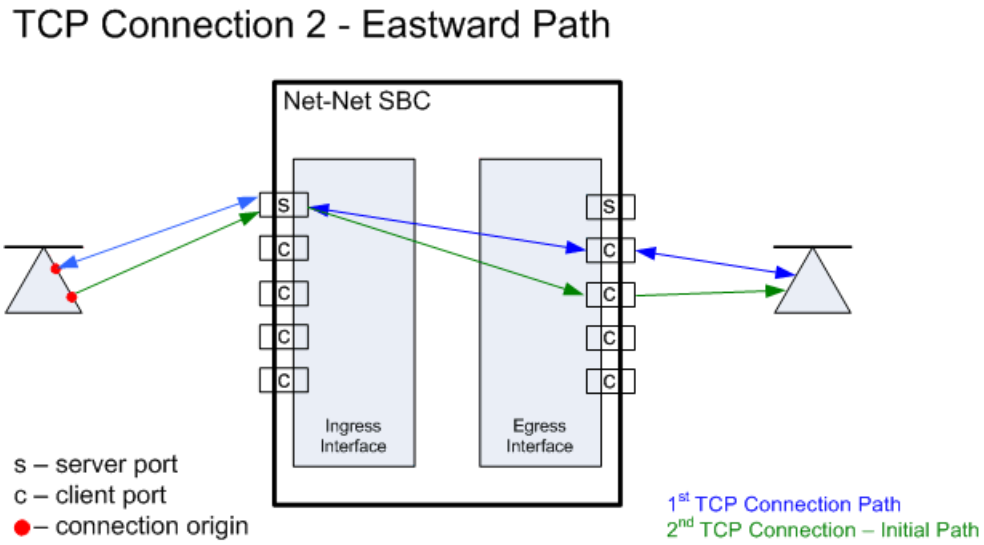
The server port is used for initiating a new TCP connection. An endpoint sends the first packet to a server port on the ingress interface. The packet is forwarded out of the Oracle Enterprise Session Border Controller through a client port on the egress interface toward an endpoint:



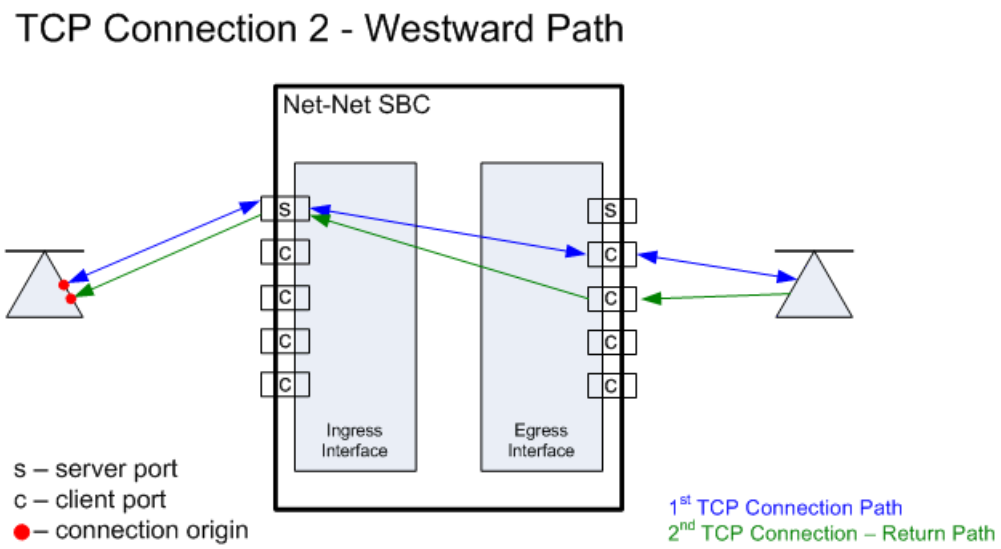
The endpoint responds back to the client port on the egress interface. This message traverses the Oracle Enterprise Session Border Controller and is forwarded out of the server port on the ingress interface where the initial packet was sent. The remainder of the TCP flow uses the server and client port pair as a tunnel through the Oracle Enterprise Session Border Controller:



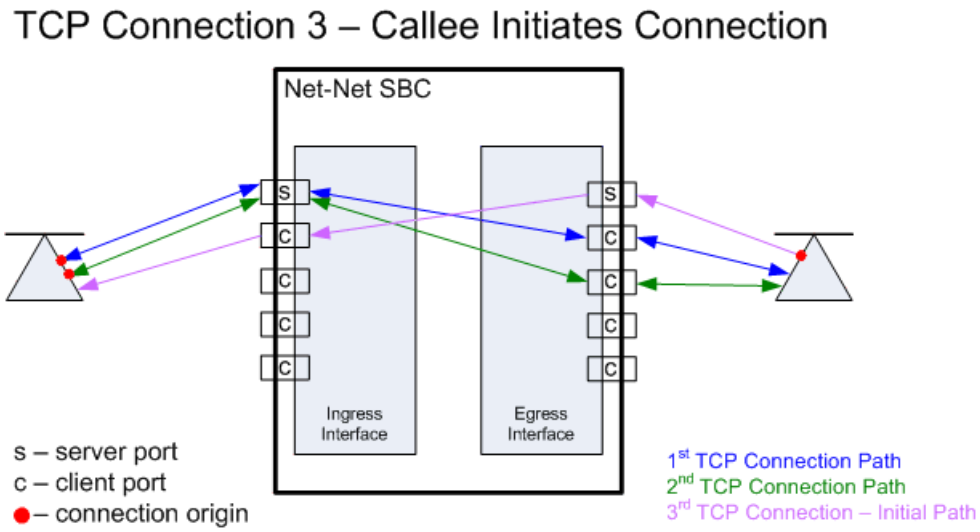
When the second TCP connection is set up in the same direction as in the first example, the first packet is still received on the server port of the ingress interface. The next unused client port is chosen for the packet to exit the Oracle Enterprise Session Border Controller:



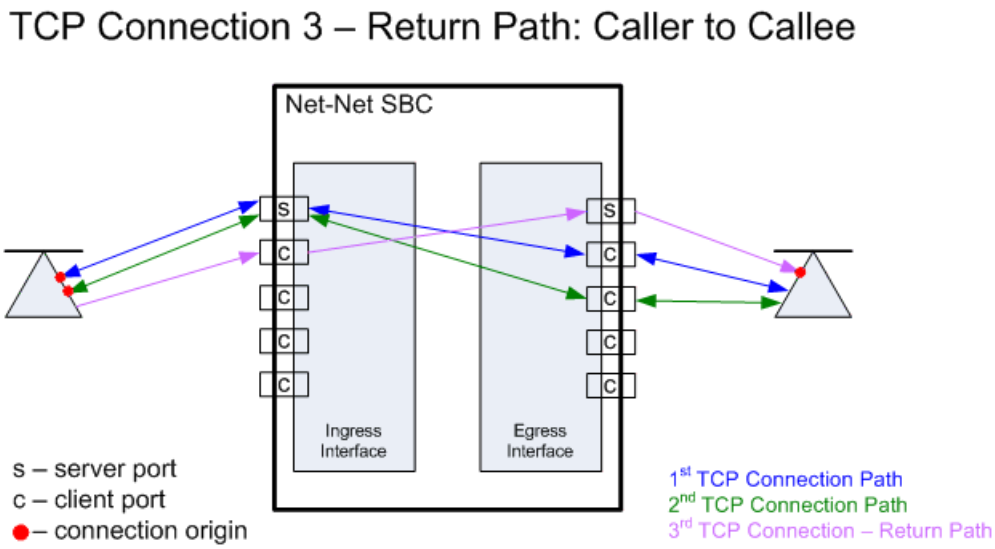
The response takes the same path back to the caller. The remainder of the second TCP connection uses this established path:



When the callee initiates a TCP connection, it must send its initial traffic to the server port on its Oracle Enterprise Session Border Controller ingress interface. The packet is forwarded out of the first free client port on the egress side of this TCP connection toward the caller.



The caller's response takes the same path back to the callee that initiated this TCP connection. The remainder of the third TCP connection uses this established path.

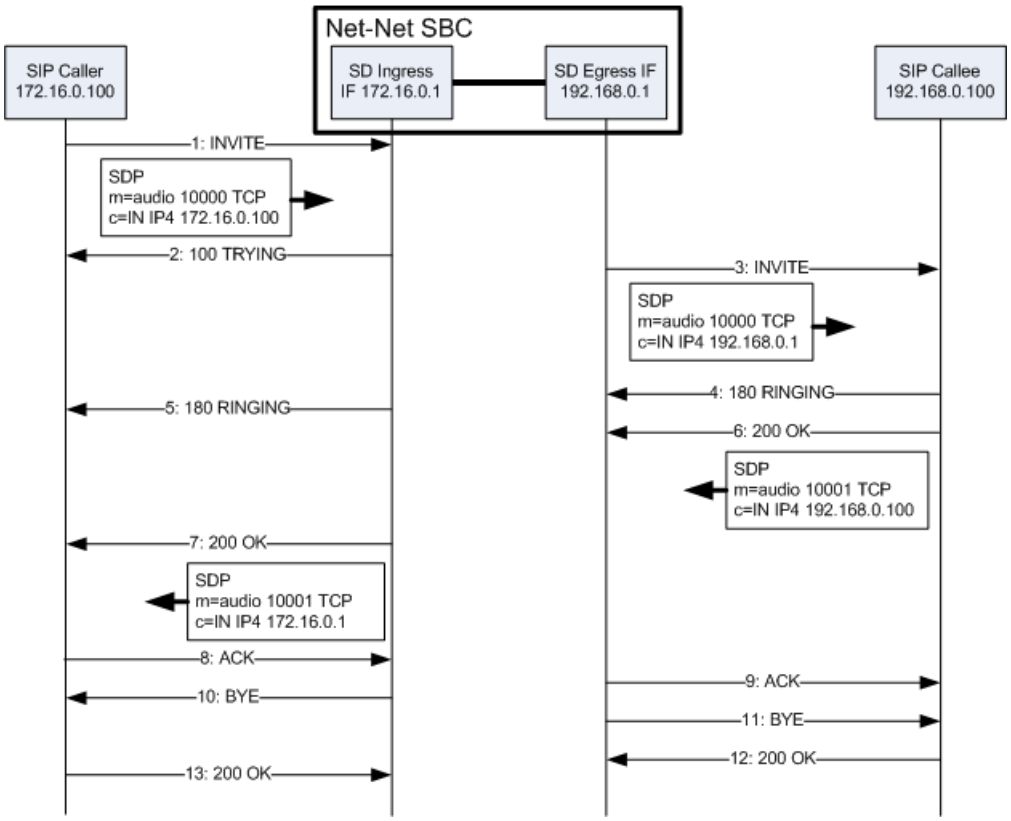


The Oracle Enterprise Session Border Controller can support a total of eight media-over-TCP connections per call. A maximum of 4 connections are supported as initiated from each side of the call.

SDP Offer Example

The following abbreviated call flow diagram sets up a media-over-TCP flow. Observe that the caller listens for audio over TCP on 172.16.0.10:10000, as described in the SDP offer (1). The Oracle Enterprise Session Border Controller re-writes the m and c lines in the SDP offer to reflect that it is listening for audio over TCP on its egress interface at 192.168.0.1:10000 (3). The Oracle Enterprise Session Border Controller then forwards the SIP invite to the callee.

The SIP callee responds with an SDP answer in a 200 OK message. The callee indicates it is listening for the audio over TCP media on 192.168.0.10:10001 (6). The Oracle Enterprise Session Border Controller re-writes the m and c lines in the SDP answer to reflect that it is listening for audio over TCP on the call's ingress interface at 172.16.0.1:10001 (7). The Oracle Enterprise Session Border Controller then forwards the SIP invite to the caller.



All interfaces involved with the end-to-end TCP flow have now established their listening IP address and port pairs.

Timers

The Oracle Enterprise Session Border Controller has three guard timers that ensure a TCP media flow does not remain connected longer than configured. You can set each of these from 0 (disabled) to 999999999 in seconds.

- TCP initial guard timer — Sets the maximum time in seconds allowed to elapse between the initial SYN packet and the next packet in this flow.
- TCP subsequent guard timer — Sets the maximum time in seconds allowed to elapse between all subsequent sequential TCP packets.
- TCP flow time limit — Sets the maximum time that a single TCP flow can last. This does not refer to the entire call.

TCP Port Configuration

To configure media over TCP:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type media-manager and press Enter to begin configuring media over TCP.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. tcp-number-of-ports-per-flow—Enter the number of ports, inclusive of the server port, to use for media over TCP. The total number of supported flows is this value minus one. The default is 2. The valid range is:

   - Minimum—2

- Maximum—5

```
ACMEPACKET(realm-config)# tcp-number-of-ports-per-flow 5
```

5. tcp-initial-guard-timer—Enter the maximum time in seconds allowed to elapse between the initial SYN packet and the next packet in a media-over-TCP flow. The default is 300. The valid range is:

   - Minimum—0
   - Maximum—999999999

   ```
   ACMEPACKET(realm-config)# tcp-initial-guard-timer 300
   ```

6. tcp-subsq-guard-timer—Enter the maximum time in seconds allowed to elapse between all subsequent sequential media-over-TPC packets. The default is 300.

   - Minimum—0
   - Maximum—999999999

   ```
   ACMEPACKET(realm-config)# tcp-subsq-guard-timer 300
   ```

7. tcp-flow-time-limit—Enter the maximum time in seconds that a media-over-TCP flow can last. The default is 86400. The valid range is:

   - Minimum—0
   - Maximum—999999999

   ```
   ACMEPACKET(realm-config)# tcp-flow-time-limit 86400
   ```

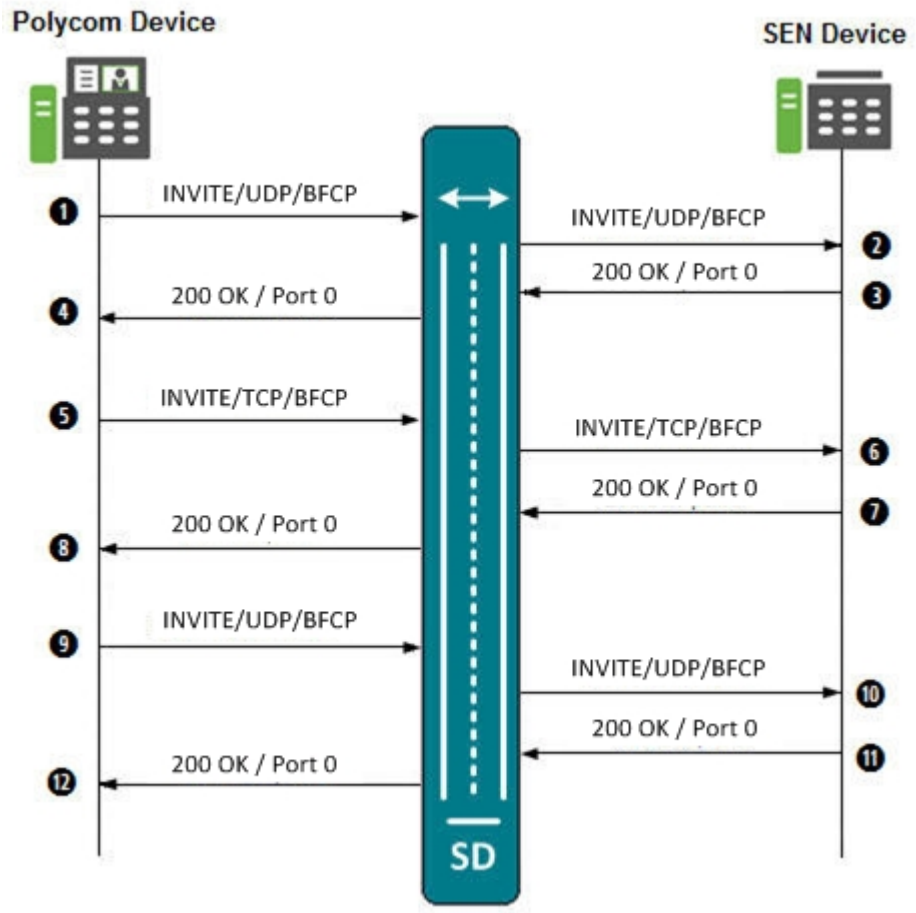
Transparent BFCP Support over UDP and TCP

Binary Floor Control Protocol (BFCP) is a protocol for controlling the access to the media resources in a conference, such as conference and media session setup, conference policy manipulation, and media control (as defined in RFC 4582).

The Oracle Enterprise Session Border Controller now supports BFCP for interworking between Polycom video devices and Siemens Enterprise Communications (SEN) endpoints. When a SIP INVITE request containing a Session Description Protocol (SDP) from a Polycom device is sent to a SEN device, the Oracle Enterprise Session Border Controller passes the INVITE request between the two devices regardless of the transfer protocol being used by the devices (UDP or TCP). It also passes the INVITE whether or not it is accepted or rejected by the destination device. The transfer protocol changes between UDP and TCP during the dialog between both endpoints on either side of the Oracle Enterprise Session Border Controller.

Note: If both endpoints on either side of the Oracle Enterprise Session Border Controller support BFCP, the BFCP is answered with the first SDP offer/answer cycle.

The following illustrates the call flow between a Polycom device and a SEN device when an INVITE is sent from the Polycom device.



The following table describes the call flow process.

Call Flow Description	
① Polycom device initiates a call to the SEN device by sending a SIP INVITE to the SD with SDP, using UDP and BFCP.	⑦ SEN device does not support BFCP, and therefore, rejects the re-INVITE, and sends a 200 Ok with port ‘0’ from the SEN side to the SD.
② SD forwards the SIP INVITE to the SEN device.	⑧ SD forwards the 200 Ok response to the Polycom device.
③ SEN device does not support BFCP, and therefore, rejects the INVITE and sends a 200 Ok with port '0' from the SEN side to the SD.	⑨ Polycom device looks at port ‘0’ and changes the media transport type from TCP to UDP. It then sends a re-INVITE to the SD.
④ SD forwards the 200 Ok response to the Polycom device.	⑩ SD forwards the re-INVITE to the SEN device.
⑤ Polycom device looks at port ‘0’ and changes the media transport type from UDP to TCP. It then sends a re-INVITE to the SD.	⑪ SEN device does not support BFCP, and therefore, rejects the re-INVITE, and sends a 200 Ok with port ‘0’ from the SEN side to the SD.
⑥ SD forwards the re-INVITE to the SEN device.	⑫ SD forwards the 200 Ok response to the Polycom device.
	Process repeats Steps 5 through 12 until the call is accepted by the SEN device.

Restricted Media Latching

The restricted media latching feature lets the Oracle Enterprise Session Border Controller latch only to media from a known source IP address, in order to learn and latch the dynamic UDP port number. The restricting IP address's origin can be either the SDP information or the SIP message's Layer 3 (L3) IP address, depending on the configuration.

About Latching

Latching is when the Oracle Enterprise Session Border Controller listens for the first RTP packet from any source address/port for the destination address/port of the Oracle Enterprise Session Border Controller. The destination address/port is allocated dynamically and sent in the SDP. After it receives a RTP packet for that allocated destination address/port, the Oracle Enterprise Session Border Controller only allows subsequent RTP packets from that same source address/port for that particular Oracle Enterprise Session Border Controller destination address/port. Latching does not imply that the latched source address/port is used for the destination of the reverse direction RTP packet flow (it does not imply the Oracle Enterprise Session Border Controller will perform symmetric RTP).

Restricted Latching

The Oracle Enterprise Session Border Controller restricts latching of RTP/RTCP media for all calls within a realm. It latches to media based on one of the following:

- SDP: the IP address and address range based on the received SDP c= connect address line in the offer and answer.
- Layer 3: the IP address and address range based on the received L3 IP address of the offer or answer. This option is for access registered HNT endpoints. If the L3 IP address is locally known and cached by the Oracle Enterprise Session Border Controller as the public SIP contact address, that information could be used instead of waiting for a response. The Oracle Enterprise Session Border Controller might use the L3 IP address restriction method for all calls regardless of whether the endpoint is behind a NAT or not, for the same realms.

Symmetric Latching

A mode where a device's source address/ports for the RTP/RTCP it sends to the Oracle Enterprise Session Border Controller (E-SBC) that are latched, are then used for the destination of RTP/RTCP sent to the device.

After allocating the media session in SIP, the E-SBC sets the restriction mode and the restriction mask for the calling side as well as for the called side. It sets the source address and address prefix bits in the flow. It also parses and loads the source flow address into the MIBOCO messages. After receiving the calling SDP, the E-SBC sets the source address (address and address prefix) in the appropriate flow (the flow going from calling side to the called side). After receiving the SDP from the called side, the E-SBC sets the source address in the flow going from the called side to the calling side.

The E-SBC uses either the address provided in the SDP or the layer 3 signaling address for latching. You also configure the E-SBC to enable latching so that when it receives the source flow address, it sets the address and prefix in the NAT flow. When the NAT entry is installed, all the values are set correctly. In addition, sipd sends the information for both the incoming and outgoing flows. After receiving SDP from the called side sipd, the E-SBC sends information for both flows to the MBCD so that the correct NAT entries are installed.

Enabling restricted latching may make the E-SBC wait for a SIP/SDP response before latching, if the answerer is in a restricted latching realm. This is necessary because the E-SBC does not usually know what to restrict latching to until the media endpoint is reached. The only exception could be when the endpoint's contact/IP is cached.

Relationship to Symmetric Latching

The current forced HNT symmetric latching feature lets the Oracle Enterprise Session Border Controller assume devices are behind NATs, regardless of their signaled IP/SIP/SDP layer addresses. The Oracle Enterprise Session Border Controller latches on any received RTP destined for the specific IP address/port of the Oracle Enterprise Session Border Controller for the call, and uses the latched source address/port for the reverse flow destination information.

If both restricted latching and symmetric latching are enabled, the Oracle Enterprise Session Border Controller only latches if the source matches the restriction, and the reverse flow will only go to the address/port latched to, and thus the reverse flow will only go to an address of the same restriction.

- Symmetric latching is enabled.

  If symmetric latching is enabled, the Oracle Enterprise Session Border Controller sends the media in the opposite direction to the same IP and port, after it latches to the source address of the media packet.
- Symmetric latching is disabled.

  If symmetric latching is disabled, the Oracle Enterprise Session Border Controller only latches the incoming source. The destination of the media in the reverse direction is controlled by the SDP address.

Example 1

A typical example is when the Oracle Enterprise Session Border Controller performs HNT and non-HNT registration access for endpoints. Possibly the SDP might not be correct, specifically if the device is behind a NAT. Therefore the Oracle Enterprise Session Border Controller needs to learn the address for which to restrict the media latching, based on the L3 IP address. If the endpoint is not behind a NAT, then the SDP could be used instead if preferred. However, one can make some assumptions that access-type cases will require registration caching, and the cached fixed contact (the public FW address) could be used instead of waiting for any SDP response.

Example 2

Another example is when a VoIP service is provided using symmetric-latching. A B2BUA/proxy sits between HNT endpoints and the Oracle Enterprise Session Border Controller, and calls do not appear to be behind NATs from the Oracle Enterprise Session Border Controller's perspective. The Oracle Enterprise Session Border Controller's primary role, other than securing softswitches and media gateways, is to provide symmetric latching so that HNT media will work from the endpoints.

To ensure the Oracle Enterprise Session Border Controller's latching mechanism is restricted to the media from the endpoints when the SIP Via and Contact headers are the B2BUA/proxy addresses and not the endpoints', the endpoint's real (public) IP address in the SDP of the offer/answer is used. The B2BUA/proxy corrects the c= line of SDP to that of the endpoints' public FW address.

The Oracle Enterprise Session Border Controller would then restrict the latching to the address in the SDP of the offer from the access realm (for inbound calls) or the SDP answer (for outbound calls).

Restricted Latching Configuration

To configure restricted latching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>           0.0.0.0
2: MGCP_Realm <none>           0.0.0.0
3: H323REALM  <none>           0.0.0.0
selection:1
ACMEPACKET(realm-config)#
```

5. restricted-latching— Enter the restricted latching mode. The default is none. The valid values are:
   - none—No restricted-latching used
   - sdp—Use the address provided in the SDP for latching
   - peer-ip—Use the layer 3 signaling address for latching
6. restriction-mask— Enter the number of address bits you want used for the source latched address. This field will be used only if the restricted-latching is used. The default is 32. When this value is used, the complete IP address is matched for IPv4 addresses. The valid range is:
   - Minimum—1
   - Maximum—128

Media Release Across SIP Network Interfaces

This feature lets the Oracle Enterprise Session Border Controller release media between two SIP peers, between two realms on two network interfaces of the same Oracle Enterprise Session Border Controller. Use this feature when you want the Oracle Enterprise Session Border Controller to release media for specific call flows, regardless of the attached media topology.

Media Release Configuration

To configure media release across network interfaces:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>           0.0.0.0
2: MGCP_Realm <none>           0.0.0.0
3: H323REALM  <none>           0.0.0.0
selection:1
ACMEPACKET(realm-config)#
```

5. mm-in-system—Set this parameter to enabled to manage/latch/steer media in the Oracle Enterprise Session Border Controller. Set this parameter to disabled to release media in the Oracle Enterprise Session Border Controller.

   Note: Setting this parameter to disabled will cause the Oracle Enterprise Session Border Controller to NOT steer media through the system (no media flowing through this Oracle Enterprise Session Border Controller).

   The default is enabled. The valid values are:

   - enabled | disabled

Media Release Behind the Same IP Address

The media management behind the same IP feature lets the Oracle Enterprise Session Border Controller release media when two endpoints are behind the same IP address, in the same realm. Using this feature prevents the media for intra-site calls from going through the Oracle Enterprise Session Border Controller. You can use this feature for both hosted NAT traversal (HNT) and non-HNT clients. It works with NATed endpoints and for non-NATed ones that are behind the same IP.

Additional Media Management Options

Additional media management options include:

- Media directed between sources and destinations within this realm on this specific Oracle Enterprise Session Border Controller. Media travels through the Oracle Enterprise Session Border Controller rather than straight between the endpoints.
- Media directed through the Oracle Enterprise Session Border Controller between endpoints that are in different realms, but share the same subnet.
- For SIP only, media released between multiple Oracle Enterprise Session Border Controllers.

  To enable SIP distributed media release, you must set the appropriate parameter in the realm configuration. You must also set the SIP options parameter to media-release with the appropriate header name and header parameter information. This option defines how the Oracle Enterprise Session Border Controller encodes IPv4 address and port information for media streams described by, for example, SDP.

Configuring Media Release Behind the Same IP Address

You need to configure both the mm-in-realm and mm-same-ip parameters for the realm:

- If the mm-in-realm parameter is disabled, the mm-same-ip parameter is ignored.
- If the mm-in-realm parameter is enabled and the mm-same-ip parameter is disabled, media will be managed in the realm but released if the two endpoints are behind the same IP address.

To configure media management:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

   From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

4. mm-in-realm—Enable if you plan to use mm-same-ip. If this parameter is disabled, the mm-same-ip parameter is ignored. If you set this to enabled and mm-same-ip to disabled, media is managed in the realm but released if the two endpoints are behind the same IP address. The default is disabled. The valid values are:

   - enabled | disabled

5. mm-same-ip—Enable if you want media to go through this Oracle Enterprise Session Border Controller, if mm-in-realm is enabled. When disabled, the media will not go through the Oracle Enterprise Session Border Controller for endpoint that are behind the same IP. The default is enabled. The valid values are:

   - enabled | disabled

Bandwidth CAC for Media Release

The bandwidth CAC for media release feature adds per-realm configuration that determines whether or not to include inter-realm calls in bandwidth calculations. When you use this feature, the Oracle Enterprise Session Border Controller's behavior is to count and subtract bandwidth from the used bandwidth for a realm when a call within a single site has its media released. When you do not enable this feature (and the Oracle Enterprise Session Border Controller's previous behavior), the Oracle Enterprise Session Border Controller does not subtract the amount of bandwidth.

In other words:

- When you enable this feature, an inter-realm media-released call will decrement the maximum bandwidth allowed in that realm with the bandwidth used for that call.
- When you disable this feature (default behavior), and inter-realm media-released call will not decrement the maximum bandwidth allowed for that call.

Bandwidth CAC Configuration

To enable bandwidth CAC for media release:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to want to add this feature.

```
ACMEPACKET(realm-config)# select
```

5. bw-cac-non-mm—Enable this parameter to turn on bandwidth CAC for media release. The default is disabled. The valid values are:
   - enabled | disabled
6. Save and activate your configuration.

Media Release between Endpoints with the Same IP Address

You can configure your Oracle Enterprise Session Border Controller to release media between two endpoints even when one of them:

- Is directly addressable at the same IP address as a NAT device, but is not behind a NAT device
- Is at the same IP address of a NAT device the other endpoint is behind

You enable this feature on a per-realm basis by setting an option in the realm configuration.

When this option is not set, theOracle Enterprise Session Border Controller will (when configured to do so) release media between two endpoints sharing one NAT IP address in the same realm or network.

Media Release Configuration

In order for this feature to work properly, the following conditions apply for the realm configuration:

- Either the mm-in-realm or the mm-in-network parameter must be disabled; you can have one of these enabled as long as the other is not. The new option will apply to the parameter that is disabled.

- If either the mm-in-realm or mm-in-network parameter is enabled, then the mm-same-ip parameter must be disabled.

To enable media release between endpoints with the same IP address:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. options—Set the options parameter by typing options, a Space, the option name release-media-at-same-nat with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Media Release Behind the Same NAT IP Address

You can now configure your Oracle Enterprise Session Border Controller to release media between endpoints sharing the same NAT IP address, even if one endpoint is at—but not behind—the same NAT. This feature expands on the Oracle Enterprise Session Border Controller'S pre-existing ability to release media between calling and called parties behind the same IP address/NAT device in the same realm or network.

Media Release Configuration

For this feature to work properly, your realm configuration should either have the mm-in-realm or mm-in-network parameter set to disabled, unless the mm-same-ip parameter is set to disabled. If the mm-same-ip parameter is enabled, then mm-in-realm or mm-in-network can both be enabled.

To set the option that enables media release behind the same IP address:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. options—Set the options parameter by typing options, a Space, the option name release-media-at-same-nat with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Codec Reordering

Certain carriers deploy voice services where their peering partners do not use the carriers' preferred codecs. The Oracle Enterprise Session Border Controller can now reorder the codecs so that the preferred one is selected first.

Take the example of a carrier that deploys a voice service using G.729 rather than G.711. If that carrier has a peering partner providing call origination for the VoIP customers with G.711 used as the preferred codec, there can be issues with codec selection.

The Oracle Enterprise Session Border Controller resolves this issue by offering its codec reordering feature. Enabled for realms and session agents, this feature gives the Oracle Enterprise Session Border Controller the ability to reorder the default codec in an SDP offer to the preferred codec before it forwards the offer to the target endpoint. When you enable this feature, you increase the probability that the target endpoint will choose the preferred codec for its SDP answer, thereby avoiding use of the undesired codec.

You enable codec reordering feature by setting the preferred-codec=X (where X is the preferred codec) option in the realm and session agent configurations. You set it in the realm from which the Oracle Enterprise Session Border Controller receives SDP offers (in requests or responses), and for which the media format list needs to be reordered by the Oracle Enterprise Session Border Controller prior to being forwarded. To configure additional codec ordering support for cases when a response or request with an SDP offer is from a session agent, you can set this option in the session agent configuration.

If you enable the option, the Oracle Enterprise Session Border Controller examines each SDP media description before if forwards an SDP offer. And if necessary, it performs reordering of the media format list to designate that the preferred codec as the default.

The Oracle Enterprise Session Border Controller determines preferred codecs in the following ways:

- If the response or request with an SDP offer is from a session agent, the Oracle Enterprise Session Border Controller determines the preferred codec by referring to the session agent configuration. You set the preferred codec for a session agent by configuring it with the preferred-codec=X option.
- If the response or request with an SDP offer is not from a session agent or is from a session agent that does not have the preferred-codec=X option configured, the Oracle Enterprise Session Border Controller determines the preferred codec by referring to the preferred-codec=X option in the realm.
- If the Oracle Enterprise Session Border Controller cannot determine a preferred codec, it does not perform codec reordering.

The way that the Oracle Enterprise Session Border Controller performs codec reordering is to search for the preferred codec in the SDP offer's media description (m=) line, and designate it as the default codec (if it is not the default already). After it marks the preferred codec as the default, the Oracle Enterprise Session Border Controller does not perform any operation on the remaining codecs in the media format list.

Note: that the Oracle Enterprise Session Border Controller performs codec reordering on the media format list only. If the rtpmap attribute of the preferred codec is present, the Oracle Enterprise Session Border Controller does not reorder it.

Preferred Codec Precedence

When you configure preferred codecs in session agents or realms, be aware that the codec you set for a session agent takes precedence over one you set for a realm. This means that if you set preferred codecs in both configurations, the one you set for the session agent will be used.

In the case where the Oracle Enterprise Session Border Controller does not find the session agent's preferred codec in the SDP offer's media format list, then it does not perform codec reordering even if the media format list contains the realm's preferred codec.

Codec Reordering Configuration

When you configure codec ordering, the codec you set in either the session agent or realm configuration must match the name of a media profile configuration. If your configuration does not use media profiles, then the name of the preferred codec that you set must be one of the following:

- PCMU
- G726-32
- G723
- PCMA
- G722
- G728
- G729

Note: If you configure this feature for a session agent, you must configure it for the associated realm as well. Otherwise, the feature will not work correctly.

Setting a Preferred Codec for a Realm

To set a preferred codec for a realm configuration:

These instructions assume that you want to add this feature to a realm that has already been configured.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: public     media2:0         0.0.0.0
2: private    media1:0         0.0.0.0
selection:1
ACMEPACKET(realm-config)#
```

5. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (preferred-codec=X), and then press Enter. X is the codec that you want to set as the preferred codec.

```
ACMEPACKET(realm-config)# options +preferred-codec=PCMU
```

   If you type options preferred-codec=X, you will overwrite any previously configured options. In order to append the new option to the realm-config's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

Setting a Preferred Codec for a Session Agent

To set a preferred codec for a session agent configuration:

These instructions assume that you want to add this feature to a session agent that has already been configured.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Select the session agent where you want to apply this feature.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: acmepacket.com realm=       ip=
2: sessionAgent2  realm=tester ip=172.30.1.150
selection:
selection:1
ACMEPACKET(session-agent)#
```

5. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (preferred-codec=X), and then press Enter. X is the codec that you want to set as the preferred codec.

```
ACMEPACKET(session-agent)# options +preferred-codec=PCMU
```

   If you type options preferred-codec=X, you will overwrite any previously configured options. In order to append the new option to the session agent's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

Media Profiles Per Realm

For different codecs and media types, you can set up customized media profiles that serve the following purposes:

- Police media values
- Define media bandwidth policies
- Support H.323 slow-start to fast-start interworking

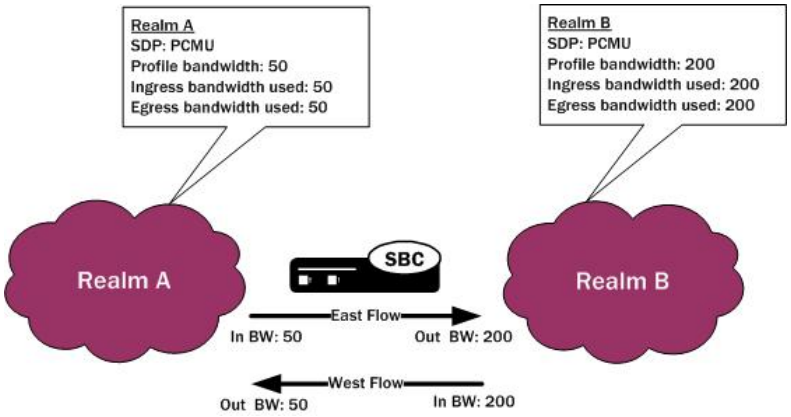
You can use media policies globally for the Oracle Enterprise Session Border Controller, or—starting with Release C6.1.0—you can configure them for application on a per-realm basis. For a realm, you can configure a list of media profiles you want applied. The Oracle Enterprise Session Border Controllermatches the value you set for the match-media-profiles parameter, and then applies those media profiles to the realm itself and to all of its child realms (but not to its parent realms).

Note: This feature has no impact on the ways the Oracle Enterprise Session Border Controller uses media profiles non-realm applications such as: H.323 interfaces, SIP interfaces, IWF, session agents, codec policies, and policy attributes.

Call Admission Control and Policing

The Oracle Enterprise Session Border Controller supports call admission control (CAC) based on realm, and it applies the limits on either ingress or egress bandwidth counters. If a calls exceeds bandwidth on either the ingress or egress side, the Oracle Enterprise Session Border Controller rejects the call. You can also use per-user CAC, which limits the maximum bandwidth from the east and west flows for both the TO and FROM users.

When you apply media profiles to a realm, the Oracle Enterprise Session Border Controller applies bandwidth policing from the flow's ingress realm media profile. In the diagram below, the Oracle Enterprise Session Border Controller policies traffic for Realm A based Realm A's policing values, and the same is true for Realm B.



Media Profile Configuration

This section shows you how to configure multiple media profiles per realm, and it explains how to use wildcarding.

To reference a media profile in this list, you need to enter its name and subname values in the following format <name>::<subname>. Releases C6.1.0 and later accept the subname so you can configure multiple media profile for the same codec; the codec name customarily serves and the name value for a media profile configuration.

About Wildcarding

You can wildcard both portions (name and subname) of this value:

- When you wildcard the name portion of the value, you can provide a specific subname that the Oracle Enterprise Session Border Controller uses to find matching media profiles.
- When you wildcard the subname portion of the value, you can provide a specific name that the Oracle Enterprise Session Border Controller uses to find matching media profiles.

You can also enter the name value on its own, or wildcard the entire value. Leaving the subname value empty is also significant in that it allows the realm to use all media profile that have no specified subname. However, you cannot leave the name portion of the value unspecified (as all media profiles are required to have names).

Consider the examples in the following table:

Syntax	Example Value	Description
<name>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <name>::*.
<name>::	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<name>::<subname>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
*	*	Matches anything, but does not have to be a defined media profile.
::	*::*	Matches any and all media profiles, but requires the presence of media profile configurations.
*::<subname>	*::64k	Matches all media profiles with this subname. You might have a group of media profiles with different names, but the same subname value.
*::	*::	Matches any media profiles with an empty subname parameter.
::	::	Invalid

Syntax	Example Value	Description
::*	::*	Invalid

The Oracle Enterprise Session Border Controller performs matching for wildcarded match-media-profiles values last. Specific entries are applies first and take precedence. When the Oracle Enterprise Session Border Controller must decide between media profiles matches, it selects the first match.

To use media profiles for a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. match-media-profiles—In the form <name>::<subname>, enter the media profiles you would like applied to this realm. These values correspond to the name and subname parameters in the media profile configuration. You can wildcard either of these portions of the value, or you can leave the <subname> portion empty.

   This parameter has no default.

5. Save and activate your configuration.

Multiple Media Profiles

You can use the media profiles configuration to set up:

- One media profile for a particular SIP SDP encoding (such as G729), where the name of the profile identifies it uniquely. This behavior is your only option in Oracle Enterprise Session Border Controller release prior to Release C6.1.0.
- Multiple media profiles for the same SIP SDP encoding. Available in Release C6.1.0 and forward, you can create multiple media profiles for the same encoding. To do so, you add a subname to the configuration, thereby identifying it uniquely using two pieces of information rather than one.

The sections below provide two descriptions of deployments where using multiple profiles for the same codec would solve codec and packetization problems for service providers.

Use Case 1

Service Provider 1 peers with various carriers, each of which uses different packetization rates for the same codec. For example, their Peer 1 uses 10 milliseconds G.711 whereas their Peer 2 uses 30 milliseconds for the same codec. The difference in rates produces a difference in bandwidth consumption—resulting in a difference in SLA agreements and in Oracle Enterprise Session Border Controller call admission control (CAC) and bandwidth policing. Service Provider 1 uses the Oracle Enterprise Session Border Controller's media profile configuration parameters to determine CAC (req-bandwidth) and bandwidth policing (avg-rate-limit). Because this service provider's peers either do not use the SDP p-time attribute or use it inconsistently, it is difficult to account for bandwidth use. And so it is likewise difficult to set up meaningful media profiles.

The best solution for this service provider—given its traffic engineering and desire for the cleanest routing and provisioning structures possible—is to define multiple media profiles for the same codec.

Use Case 2

Service Provider 2 supports H.263 video, for which the Oracle Enterprise Session Border Controller offers a pre-provisioned media profile with a set bandwidth value. And yet, H.263 is not a codec that has a single bandwidth value. Instead, H.263 can have different bandwidth values that correspond to various screen resolution and quality. While it is true that the Oracle Enterprise Session Border Controllercan learn the requisite bandwidth value from SDP, not all SDP carries the bandwidth value nor do system operators always trust the values communicated.

Configuring multiple media profiles for the same codec (here, H.263) helps considerably with this problem—and moves closer to complete solution. Service Provider 2 can configure H.263 media profiles capable of handling the different bandwidth values that might appear.

Multiple Media Profiles Configuration

Configuring the subname parameter in the media profiles configuration allows you to create multiple media profiles with the same name.

To configure the subname parameter for a media profile:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type media-profile and press Enter. If you are adding this feature to a pre-existing media profile configuration, you will need to select and edit your media profile.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. subname—Enter the subname value for this media profile. Information such as the rate or bandwidth value make convenient subname values. For example, you might set the name of the media profile as PCMU and the subname as 64k.

   This parameter is not require and has no default.

5. Save and activate your configuration.

SIP Disable Media Inactivity Timer for Calls Placed on Hold

Hardware-based media flow guard timers detect when a call has lost media while it is being relayed through the Oracle Enterprise Session Border Controller. In response, the system tears down the call.

You can configure disable-guard-timer-sendonly to disable media inactivity timers for calls placed on hold. The Oracle Enterprise Session Border Controller disableds initial and subsequent guard timers for media when the SIP or IWF call is put on hold with a 0.0.0.0 address in:

- The c=connection line
- An a=inactive attribute
- An a=sendonly attribute

It should be noted that disabling the media inactivity timers will also disable the guard timers for calls which are not necessarily on hold, but simply are one-way audio applications.

No license requirements to enable this feature.

Media Inactivity Timer Configuration

To disable the media inactivity timer for calls placed on hold:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type media-manager-config and press Enter.

```
ACMEPACKET(media-manager)# media-manager-config
ACMEPACKET(media-manager-config)#
```

4. options—Set the options parameter by typing options, a Space, the option-name disable-guard-timer-sendonly with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(media-manager-config)# options +disable-guard-timer-sendonly
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the SIP interface configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

8

SIP Signaling Services

This chapter explains how to configure the Oracle Enterprise Session Border Controller to support Session Initiation Protocol (SIP) signaling services for hosted IP services applications. SIP is a text-based application-layer signaling protocol that creates, identifies, and terminates multimedia sessions between devices.

About the Oracle Enterprise Session Border Controller and SIP

This section describes the Oracle Enterprise Session Border Controller's support of SIP. It provides the basic information you need to understand before you configure the Oracle Enterprise Session Border Controller for SIP signaling.

Types of SIP Devices

There are four types of SIP devices:

- SIP user agent (UA) is an endpoint in SIP end-to-end communication. A UA is a user agent client (UAC) when it initiates a request and waits to receive a response. A UA is a user agent server (UAS) when it receives a request and generates a response. A given UA will be a UAC or a UAS depending on whether it is initiating the request or receiving the request.
- A SIP proxy (or proxy server) is an intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server's primary role is routing. Its job is to ensure that a request is sent to another entity closer to the targeted user. A proxy interprets, and if necessary, rewrites specific parts of a request message before forwarding it.
- A SIP redirect server is a UAS that generates redirect responses to requests it receives, directing the client to contact an alternate set of targets. Unlike a proxy which forwards the request to the alternate set of targets, the redirect response tells the UAC to directly contact the alternate targets.
- A SIP registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles. Proxies and redirect servers can use the information from the location service to determine the location of the targeted user.

  A redirect server and a registrar are each a special type of UA because they act as the UAS for the requests they process.

Basic Service Models

The Oracle Enterprise Session Border Controller operates as a back-to-back user agent (B2BUA) within the following two basic service models:

- peering

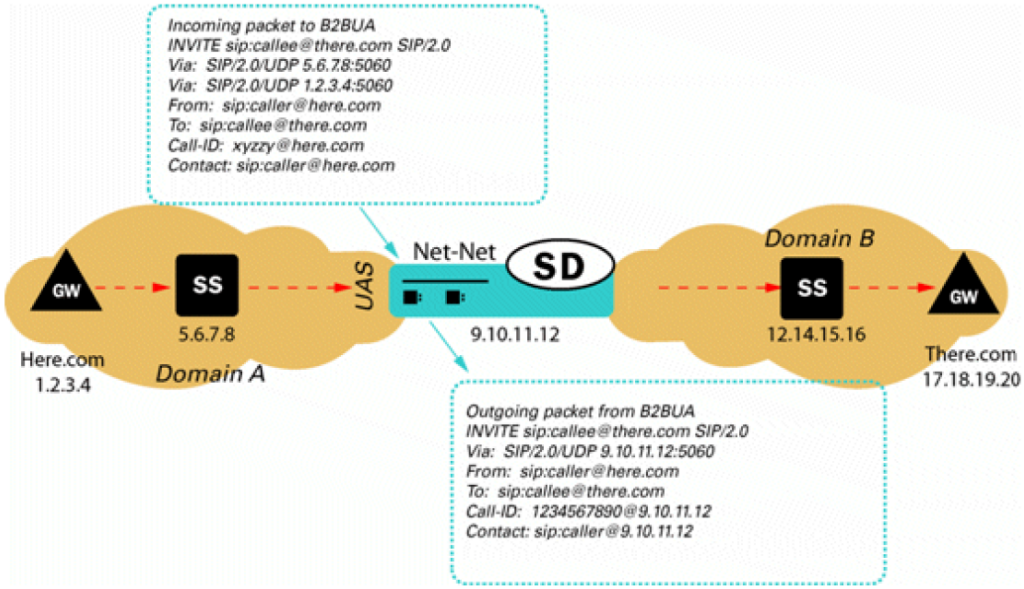
- hosted IP services

About B2BUA

A B2BUA is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests. It maintains dialog state and must participate in all requests sent on the dialogs it has established.

SIP B2BUA Peering

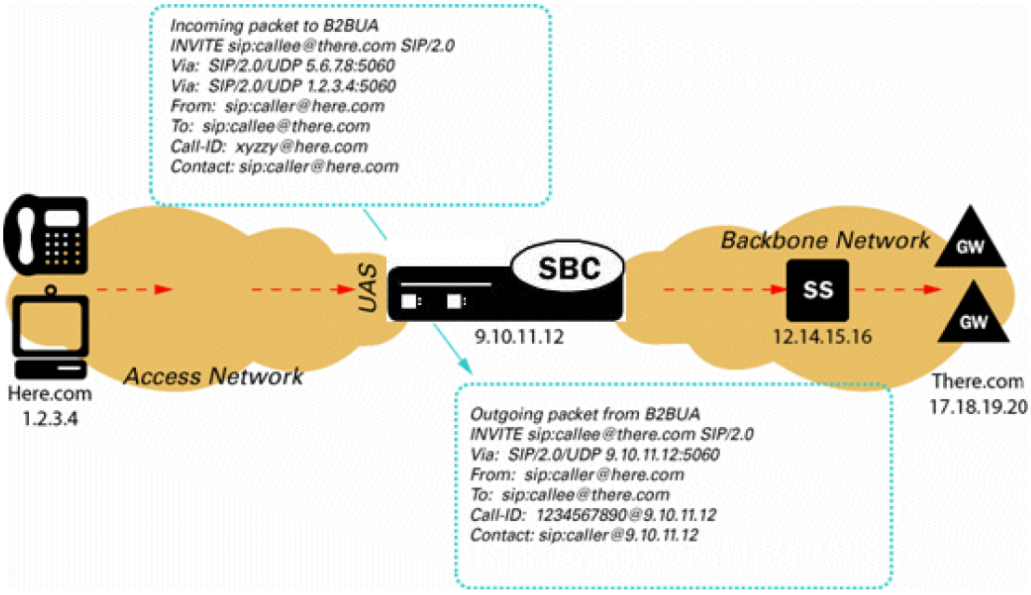
The Oracle Enterprise Session Border Controller operates as a SIP B2BUA. It terminates SIP sessions and re-originates them as new sessions as they are routed through the Oracle Enterprise Session Border Controller. For each session, it establishes NAPT translations and re-writes SDP to allow all session related media to be routed through the Oracle Enterprise Session Border Controller. It generates new call IDs and modifies SIP headers to prevent any protected SIP addresses and route information from being transmitted to external peers. The Oracle Enterprise Session Border Controller supports multiple SIP interfaces that are associated with a set of media ports, thus appearing as multiple virtual SIP gateways.



B2BUA Hosted IP Services

The Oracle Enterprise Session Border Controller acts as an outbound proxy for SIP endpoints and performs the operations required to allow UAs behind NATs to initiate and terminate SIP sessions (Hosted NAT Traversal).

The Oracle Enterprise Session Border Controller caches registration requests from SIP endpoints and forwards them to the appropriate softswitch or registrar in its backbone network. All subsequent signaling between the endpoint and the backbone network is through the Oracle Enterprise Session Border Controller. Also, all calling features such as caller ID, call waiting, three-way calling, and call transfer are all supported transparently through the Oracle Enterprise Session Border Controller.



SIP B2BUA and L3 L5 NAT

For each SIP session, the Oracle Enterprise Session Border Controller establishes NAPT translations and re-writes SDP to route all session related media through the Oracle Enterprise Session Border Controller. These actions make the Oracle Enterprise Session Border Controller look like a SIP gateway. Also, the Oracle Enterprise Session Border Controller support of multiple SIP interfaces associated with different network interfaces makes it appear as multiple virtual SIP gateways.

This functionality enables the Oracle Enterprise Session Border Controller to deliver VoIP services to multiple end users, across a VPN backbone.

About SIP Interfaces

The SIP interface defines the transport addresses (IP address and port) upon which the Oracle Enterprise Session Border Controller receives and sends SIP messages. You can define a SIP interface for each network or realm to which the Oracle Enterprise Session Border Controller is connected. SIP interfaces support both UDP and TCP transport, as well as multiple SIP ports (transport addresses). The SIP interface's SIP NAT function lets Hosted NAT Traversal (HNT) be used in any realm.

SIP INVITE Message Processing

When the session agent element on the softswitch side of the message flow (ingress session agent) has the gateway contact parameter configured as an option, the Oracle Enterprise Session Border Controller looks for the URI parameter (as defined by the gateway contact parameter) in the Request-URI and decodes the gateway address.

Example

The following example shows a SIP INVITE message from a softswitch to a Oracle Enterprise Session Border Controller.

```
INVITE sip:05030205555@ss-side-ext-address;gateway=encoded-gw-address
From: "Anonymous"<sip:anonymous@anonymous.invalid>;tag=xxxx
To: <sip:05030205555@ss-side-ext-address;user=phone>
```

The following example shows a SIP INVITE message from a Oracle Enterprise Session Border Controller to a gateway.

```
INVITE sip:05030205555@gw-ip-address SIP/2.0
From: "Anonymous"<sip:anonymous@anonymous.invalid>;tag=SDxxxx-xxxx
To: <sip:05030205555@ hostpart;user=phone>
```

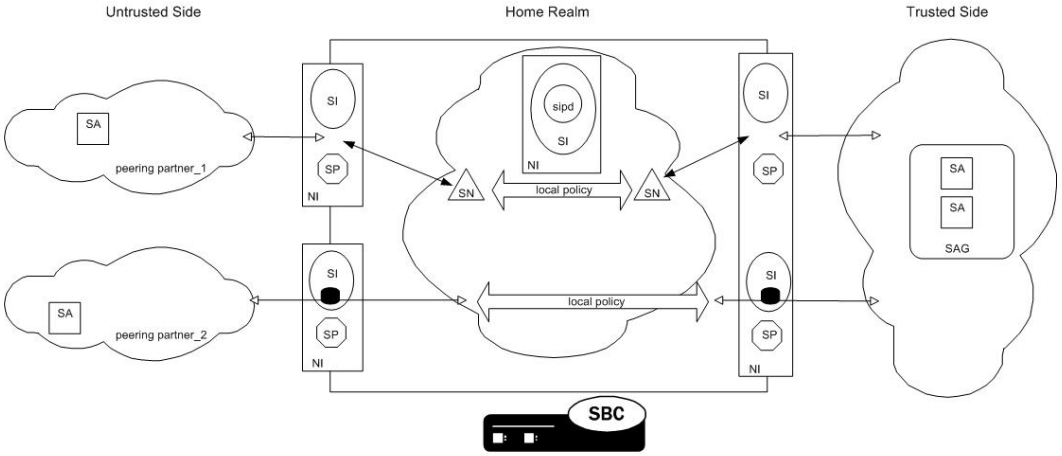
The Oracle Enterprise Session Border Controller converts the hostpart in the To header except in the following scenarios:

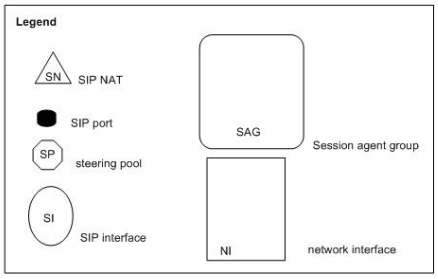
- when the original hostpart value received is an Fully Qualified Domain Name (FQDN)
- when the Oracle Enterprise Session Border Controller is configured not to NAT the To headers.

Acme Packet recommends configuring the Oracle Enterprise Session Border Controller to NAT the To headers to ensure the security of protected addresses. Otherwise, the outgoing hostpart is set to the SIP NAT's external proxy address for the SIP NAT's external realm.

Configuring the Oracle Enterprise Session Border Controller for SIP Signaling

This section contains a diagram of a B2BUA peering environment that illustrates the Oracle Enterprise Session Border Controller components you need to configure.





Home Realm

This section explains how to configure a home realm. The home realm applies only to a SIP configuration. It represents the internal default realm or network for the Oracle Enterprise Session Border Controller and is where the Oracle Enterprise Session Border Controller's SIP proxy is located.

Overview

You primarily use a home realm when using the SIP NAT function to connect multiple realms/networks to the Oracle Enterprise Session Border Controller. You define the home realm defined as either public or private for the purposes of using the SIP NAT function. If the home realm is public, all external realms are considered private. If the home realm is private, all external networks are considered public. Usually the home realm is public.

Messages are encoded (for example, the topology is hidden) when they pass from a private to a public realm. Messages are decoded when the pass from a public realm to a private realm.

These external realms/networks might have overlapping address spaces. Because SIP messages contain IP addresses, but no layer 2 identification (such as a VLAN tag), the SIP proxy must use a single global address space to prevent confusing duplicate IP addresses in SIP URIs from different realms.

SIP NAT Function

The SIP NAT function converts external addresses in SIP URIs to an internal home realm address. Usually the external address is encoded into a cookie that is added to the userinfo portion of the URI and the external address is replaced with a home realm address unique to the SIP NAT (the SIP NAT home address).

URIs are encoded when they pass from a private realm to a public realm. When an encoded URI passes back to the realm where it originated, it is decoded (the original userinfo and host address are restored). The encoding/decoding process prevents the confusion of duplicate addresses from overlapping private addresses. It can also be used to hide the private address when a SIP message is traversing a public network. Hiding the address occurs when it is a private address; or when the owner of the private network does not want the IP addresses of their equipment exposed on a public network or on other private networks to which the Oracle Enterprise Session Border Controller connects.

Home Realm's Purpose

A home realm is required because the home address for SIP NATs is used to create a unique encoding of SIP NAT cookies. You can define the home realm as a network internal to the Oracle Enterprise Session Border Controller, which eliminates the need for an actual home network connected to the Oracle Enterprise Session Border Controller. You can define this virtual home network if the supply of IP addresses is limited (because each SIP NAT requires a unique home address), or if all networks to which the Oracle Enterprise Session Border Controller is connected must be private to hide addresses.

For example, you can define a public home realm using the loopback network (127.0.0.0) and using the home realm address prefix (for example, 127.0.0.0/8) for encoding addresses that do not match (all addresses outside 127.0.0.0/8) in SIP NAT cookies. The SIP NAT address prefix field can be used to accomplish this while keeping the ability to define an address prefix for the ream for ingress realm determination and admission control. By defining the SIP NAT address prefix as 0.0.0.0, the home realm address prefix is used to encode addresses that do not match.

Home Realm Configuration

To configure the home realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP configuration parameters. To view all sip-config parameters, enter a ? at the system prompt.

4. home-realm-id—Enter the name of the realm you want to use for the realm ID. For example, acme.

   The name of the realm must correspond to the identifier value you entered when you configured the realm.

5. egress-realm-id—Optional. Enter the egress realm ID to define the default route for SIP requests addressed to destinations outside the home realm's address prefix.

   If you enter a value for this optional field, it must correspond to the identifier value you entered when you configured the realm.

   Note: You should leave this parameter blank for access/backbone applications. When left blank, the realm specified in the home-realm-id parameter is used by default.

6. nat-mode—Indicate the SIP NAT mode. The default is none. The valid values are:

   - public—Indicates the subnet defined in the addr-prefix-id field of the home realm is public and the subnet defined in the addr-prefix-id field of all external realms identified in the SIP NAT are private networks. IPv4 addresses are encoded in SIP messages received from the external realm defined by the SIP NAT. The IPv4 addresses are decoded in messages that are sent to the realm.
   - private—Indicates the subnet defined in the addr-prefix-id field of the home realm is private and the subnet defined in the addr-prefix-id field of all external realms identified in the SIP NAT are public networks. IPv4 addresses are encoded in SIP messages sent to the external realm defined by the SIP NAT and decoded in messages received from the realm.
   - none—No SIP NAT function is necessary.

   The following example shows the SIP home realm configured for a peering network.

```
sip-config
        state                           enabled
        operation-mode                  dialog
dialog-transparency        disabled
        home-realm-id                   acme
        egress-realm-id
        nat-mode                        Public
        registrar-domain
        registrar-host
        registrar-port                  0
        init-timer                      500
        max-timer                       4000
        trans-expire                    32
        invite-expire                   180
        inactive-dynamic-conn           32
        red-sip-port                    1988
        red-max-trans                   10000
        red-sync-start-time             5000
        red-sync-comp-time              1000
        last-modified-date              2005-03-19 12:41:28
```

SIP Interfaces

This section explains how to configure a SIP interface. The SIP interface defines the transport addresses (IP address and port) upon which theOracle Enterprise Session Border Controller receives and sends SIP messages.

Overview

The SIP interface defines the signaling interface. You can define a SIP interface for each network or realm to which the Oracle Enterprise Session Border Controller is connected. SIP interfaces support both UDP and TCP transport, as well as multiple SIP ports (transport addresses). The SIP interface also lets Hosted NAT Traversal (HNT) be used in any realm.

The SIP interface configuration process involves configuring the following features:

- address and transport protocols (SIP ports)
- redirect action
- proxy mode
- trust mode

About SIP Ports

A SIP port defines the transport address and protocol the Oracle Enterprise Session Border Controller will use for a SIP interface for the realm. A SIP interface will have one or more SIP ports to define the IP address and port upon which the Oracle Enterprise Session Border Controller will send and receive messages. For TCP, it defines the

address and port upon which the Oracle Enterprise Session Border Controller will listen for inbound TCP connections for a specific realm.

You need to define at least one SIP port, on which the SIP proxy will listen for connections. If using both UDP and TCP, you must configure more than one port. For example, if a call is sent to the Oracle Enterprise Session Border Controller using TCP, which it needs to send out as UDP, two SIP ports are needed.

Preferred SIP Port

When a SIP interface contains multiple SIP ports of the same transport protocol, a preferred SIP port for each transport protocol is selected for outgoing requests when the specific SIP port cannot be determined. When forwarding a request that matched a cached registration entry (HNT or normal registration caching), the SIP port upon which the original REGISTER message arrived is used. Otherwise, the preferred SIP port for the selected transport protocol is used. When selecting the preferred SIP port, the default SIP port of 5060 will be selected over other non-default ports.

For SIP interfaces using the SIP NAT function, the preferred SIP port address and port will take precedence over the external address of the SIP NAT when they do not match. If both TCP and UDP SIP ports are defined, the address and port of the preferred UDP port is used.

Proxy Mode

The Oracle Enterprise Session Border Controller's proxy mode determines whether it forwards requests received on the SIP interface to target(s) selected from local policy; or sends a send a redirect response to the previous hop. Sending the redirect response causes the previous hop to contact the targets directly.

If the source of the request matches a session agent with a proxy mode already defined, that mode overrides the proxy mode defined in the SIP interface.

You can configure the proxy mode to use the Record-Route option. Requests for stateless and transaction operation modes are forwarded with a Record-Route header that has the Oracle Enterprise Session Border Controller's addresses added. As a result, all subsequent requests are routed through the Oracle Enterprise Session Border Controller.

Redirect Action

The redirect action is the action the SIP proxy takes when it receives a SIP Redirect (3xx) response on the SIP interface. If the target of the request is a session agent with redirect action defined, its redirect action overrides the SIP interface's.

You can set the Oracle Enterprise Session Border Controllerto perform a global redirect action in response to Redirect messages. Or you can retain the default behavior where the Oracle Enterprise Session Border Controller sends SIP Redirect responses back to the previous hop (proxy back to the UAC) when the UAS is not a session agent.

The default behavior of the Oracle Enterprise Session Border Controller is to recurse on SIP Redirect responses received from the user agent server (UAS) and send a new request to the Contact headers contained in the SIP Redirect response.

Instead of this default behavior, the Oracle Enterprise Session Border Controller can proxy the SIP Redirect response back to the user agent client (UAC) using the value in the session agent's redirect action field (when the UAS is a session agent). If there are too many UASes to define as individual session agents or if the UASs are HNT endpoints, and SIP Redirect responses need to be proxied for UASs that are not session agents; you can set the default behavior at the SIP Interface level.

SIP maddr Resolution

Release S-C6.2.0 provides enhanced resolution of addresses found in SIP contact headers, or in the maddr (multicast address) parameter of SIP 3xx REDIRECT messages. Previous releases resolved these addresses as either a host address or as a session agent name. With Release 6.2.0 these addresses can also be resolved as session agent group (SAG) names.

Support for SAG-based resolution is provide by a new sip-config parameter, sag-lookup-on-redirect. By default, SAG lookup is disabled, providing compatibility with prior releases.

The following sample SIP REDIRECT and ACLI configuration fragment illustrate enhanced processing.

```
Status-Line: SIP/2.0 302 Moved
Message Header
Via: SIP/2.0/UDP
192.168.200.224:5060;branch=z9hG4bKa0fs40009o90sc8oo780.1
From: <sip:1111@192.168.1.222:6000>;tag=1
To: sut <sip:2223@192.168.1.224:5060>;tag=11
Call-ID: 1-28515@192.168.1.222
CSeq: 1 INVITE
Contact: <sip:1111@192.168.1.223;maddr=test.acmepacket.com>
Privacy: user;id;critical;session
P-Preferred-Identity: sipp <sip:sipp@192.168.200.222:5060>
P-Asserted-Identity: abc.com
Subject: abc
Proxy-Require: privacy,prack,abc
Content-Length: 0

session-group
        group-name                      test.acmepacket.com
        description
        state                           enabled
        app-protocol                    SIP
        strategy                        Hunt
        dest
                                        192.168.200.222
                                        192.168.200.223
...
...
```

In this case, when the Oracle Enterprise Session Border Controller receives the 302, it resolves the information from maddr to a SAG name. In the above example, it will resolve to the configured SAG – test.acmepacket.com. The destinations configured in SAG test.acmepacket.com will be used to route the call.

SAG-based address resolution is based on the following set of processing rules.

1. When the Contact URI does not have an maddr parameter, and the hostname is not an IP Address, the Oracle Enterprise Session Border Controller will look for a SAG matching the hostname.
2. When the Contact URI has an maddr parameter that contains an IP address, the Oracle Enterprise Session Border Controller will not look for a SAG; it will use the IP Address as the target/next-hop.
3. When the Contact URI has an maddr parameter that contains a non-IP-address value, the Oracle Enterprise Session Border Controller will look for a SAG matching the maddr parameter value.

   The above logic can be turned on by enabling sag-lookup-on-redirect in the sip-config object as shown below.

SIP maddr Resolution Configuration

To configure the Oracle Enterprise Session Border Controller to perform SAG-based maddr resolution:

1. From superuser mode, use the following command sequence to access sip-config configuration mode. While in this mode, you configure SAG-based address resolution.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```
2. Use the sag-lookup-on-redirect parameter to enable SAG-based maddr resolution.
3. Use done, exit, and verify-config to complete SAG-based address resolution.

Trust Mode

The Oracle Enterprise Session Border Controller supports the Calling Identity privacy requirements based on RFC 3323 and RFC 3325. The trust mode in the SIP interface determines whether the source and destination of a request is a trusted entity. With the implementation of this feature, the Oracle Enterprise Session Border Controller can understand and support the privacy headers and provide the capability for anonymous packets

The Oracle Enterprise Session Border Controller, which acts as a boundary device between the trusted platform and the untrusted Internet, understands the following headers:

- Privacy Header
- P-Asserted-Identity Header
- P-Preferred-Identity Header

Depending on the value of these headers and the mode in which the Oracle Enterprise Session Border Controller is being operated (B2BUA or the proxy), the appropriate actions are performed.

About the Process

On receiving a message, the Oracle Enterprise Session Border Controller checks whether the message source is trusted or not. It checks the SIP interface's trust mode value and, if the source is a session agent, the session agent's trust me value. Depending on these values, the Oracle Enterprise Session Border Controller decides whether the request's or response's source is trusted. If it receives message from a trusted source and the message contains the P-Asserted-Identity header field, the Oracle Enterprise Session Border Controller passes this message to the outgoing side. The outgoing side then decides what needs to be done with this request or response.

If the request or the response is received from an untrusted source, the Privacy header value is id (privacy is requested), and the P-Asserted-Identity header field is included, the Oracle Enterprise Session Border Controller strips the Privacy and the P-Asserted-Identity headers and passes the request or the response to the outgoing side.

If the request or the response contains the P-Preferred-Identity header and the message source is untrusted, the Oracle Enterprise Session Border Controller strips the P-Preferred-Identity header from the request or the response and passes the message to the outgoing side.

If the source is trusted or privacy is not requested (the value of the Privacy Header is not id) and the request or the response contains the P-Preferred-Identity header, the Oracle Enterprise Session Border Controller performs the following actions:

- inserts the P-Asserted-Identity header field with the value taken from the P-Preferred-Identity header field
- deletes the P-Preferred-Identity header value
- passes this request or the response to the Outgoing side for the appropriate action, depending on the whether the destination is trusted or not

After the Oracle Enterprise Session Border Controller passes the request or the response to the outgoing side, it checks whether the destination is trusted by checking the SIP interface's trust mode value and the session agent's trust me value (if the destination is configured as session agent).

- The destination is trusted

  The Oracle Enterprise Session Border Controller does nothing with the request or the response and passes it to the destination. If the P_Asserted_Identity headers are present, they are passed to the session agent (if the destination is configured as session agent).
- The destination is untrusted

  The Oracle Enterprise Session Border Controller looks at the value of the Privacy header. If set to id, the Oracle Enterprise Session Border Controller removes all the P-Asserted-Identity headers (if present). It strips the Proxy-Require header if it is set to privacy. The Oracle Enterprise Session Border Controller also sets the From field of SIP header to Anonymous and strips the Privacy header.

  If the Privacy header is set to none, the Oracle Enterprise Session Border Controller does not remove the P-Asserted-Identity header fields.

  If there is no Privacy header field, the SD will not remove the P-Asserted-Identity headers.

To implement this feature, you need to configure the session agent's trust me parameter to enabled (if the message source is a session agent) and the SIP interface's trust mode to the appropriate value.

Configurable Timers and Counters

SIP timers and counters can be set in the global SIP configuration, and two can be specific for individual SIP interfaces.

You can set the expiration times for SIP messages, and you can set a counter that restricts the number of contacts that the Oracle Enterprise Session Border Controller tries when it receives a REDIRECT. These are similar to two parameters in the global SIP configuration, trans-expire and invite-expire. You can also set a parameter that defines how many contacts/routes the Oracle Enterprise Session Border Controller will attempt on redirect.

SIP Interface Configuration

To configure a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.

4. state—Enable or disable the SIP interface. The default is enabled. The valid values are:

   - enabled | disabled

5. realm-id—Enter the name of the realm to which the SIP interface is connected.
6. sip-ports—Access the sip-ports subelement.
7. carriers—Enter the list of carriers related to the SIP interface.

   Entries in this field can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! ”$ % ^ & * ( ) + - = < > ? ‘ | { } [ ] @ / \ ‘ ~ , . _ : ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carrier are valid carrier field formats

8. proxy-mode—Enter an option for the proxy mode parameter. Valid values are:

   - proxy—Forward all SIP requests to selected targets.
   - redirect—Send a SIP 3xx redirect response with the selected target(s) in the Contact header.
   - record-route—Forward requests to selected target(s) and insert a Record-Route header with the Oracle Enterprise Session Border Controller's address. For stateless and transaction mode only.

9. redirect-action—Enter the value for the redirect action. Valid values are:

   - proxy—Send the SIP request back to the previous hop.
   - recurse—Recurses on the Contacts in the response.

     The designated proxy action will apply to SIP 3xx responses received from non-session agents and to 3xx responses received from session agents without configured SIP Redirect message actions (for example, session agents without values for the redirect action field).

10. contact-mode—Set the Contact header routing mode, which determines how the contact address from a private network is formatted.

For example, whether a maddr parameter equal to the Oracle Enterprise Session Border Controller's SIP proxy needs to be added to a URI present in a Contact header.

The default is none. The valid values are:

- none—The address portion of the header becomes the public address of that private realm.
- maddr—The address portion of the header will be set to the IP address of the Oracle Enterprise Session Border Controller's B2BUA.
- strict—The contents of the Request-URI is destroyed when a Record-Route header is present.
- loose—The Record-Route header is included in a Request, which means the destination of the request is separated from the set of proxies that need to be visited along the way.

11. nat-traversal—Define the type of HNT enabled for SIP. The default is none. Valid values are:

- none—HNT function is disabled for SIP.
- rport—SIP HNT function only applies to endpoints that include the rport parameter in the Via header. HNT applies when the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address.
- always—SIP HNT applies to requests when the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address. (Even when the rport parameter is not present.)

12. nat-interval—Set the expiration time in seconds for the Oracle Enterprise Session Border Controller's cached registration entry for an HNT endpoint. The default is 30. The valid range is:

- Minimum—0
- Maximum—999999999

  Oracle recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to cause the UA to send REGISTER messages frequently enough to retain the port binding in the NAT. Retaining the binding lets inbound requests to be sent through the NAT.

13. tcp-nat-interval—Set the registration cache expiration time in seconds to use for endpoints behind a NAT device that register using TCP. On upgrade, the Oracle Enterprise Session Border Controller assigns this parameter the same value as the existing NAT interval. The default is 90. The valid range is:

- Minimum—0
- Maximum—999999999

  The Oracle Enterprise Session Border Controller uses the value you set for the TCP NAT interval as the expiration value passed back in SIP REGISTER (200 OK) responses to endpoints behind a NAT that register over TCP. The NAT interval value with which you are familiar from previous releases is used for endpoints behind a NAT that register over UDP. Requiring endpoints that register over TCP to send refresh requests as frequently as those registering over UDP puts unnecessary load on the Oracle Enterprise Session Border Controller. By adding a separate configuration for the TCP NAT interval, the load is reduced.

  For upgrade and backward compatibility with Oracle Enterprise Session Border Controller releases prior to Release 4.1, when the tcpNatInterval is not present in the XML for a SIP interface configuration, the value of the NAT interval (natInterval) is used for the TCP NAT interval as well.

14. registration-caching—Enable for use with all UAs, not just those that are behind NATs. The default is disabled. The valid values are:

- enabled | disabled

  If enabled, the Oracle Enterprise Session Border Controller caches the Contact header in the UA's REGISTER request when it is addressed to one of the following:
- Oracle Enterprise Session Border Controller
- registrar domain value
- registrar host value

The Oracle Enterprise Session Border Controller then generates a Contact header with the Oracle Enterprise Session Border Controller's address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.

Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:

- HNT enabled: the Oracle Enterprise Session Border Controller takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.
- HNT disabled: the user part of the Contact header is taken from the URI in the From header and no cookie is appended. This is the default behavior of the Oracle Enterprise Session Border Controller.

When the registrar receives a request that matches the address-of-record (the To header in the REGISTER message), it sends the matching request to the Oracle Enterprise Session Border Controller, which is the Contact address. Then, the Oracle Enterprise Session Border Controller forwards the request to the Contact-URI it cached from the original REGISTER message.

15. min-reg-expire—Set the time in seconds for the SIP interface. The value you enter here sets the minimum registration expiration time in seconds for HNT registration caching. The default is 300. The valid range is:

    - Minimum—0
    - Maximum—999999999

    This value defines the minimum expiration value the Oracle Enterprise Session Border Controller places in each REGISTER message it sends to the real registrar. In HNT, the Oracle Enterprise Session Border Controller caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

    Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the Oracle Enterprise Session Border Controller to send frequent registrations on to the real registrar.

16. registration-interval—Set the Oracle Enterprise Session Border Controller's cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the Oracle Enterprise Session Border Controller to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires. The default is 3600. The valid range is:

    - Minimum—0

      A registration interval of zero causes the Oracle Enterprise Session Border Controller to pass back the expiration time set by and returned in the registration response from the registrar.
    - Maximum—999999999

      If the expiration time you set is less than the expiration time set by and returned from the real registrar, the Oracle Enterprise Session Border Controller responds to the refresh request directly rather than forwarding it to the registrar.

      Although the registration interval applies to non-HNT registration cache entries, and the loosely related NAT interval applies to HNT registration cache entries, you can use the two in combination. Using a combination of the two means you can implement HNT and non-HNT architectures on the same Oracle Enterprise Session Border Controller. You can then define a longer interval time in the registration interval field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

17. route-to-registrar—Enable routing to the registrar to send all requests that match a cached registration to the destination defined for the registrar host; used when the Request-URI matches the registrar host value or the registrar domain value, not the Oracle Enterprise Session Border Controller's address. Because the registrar host is the real registrar, it should send the requests back to the Oracle Enterprise Session Border Controller with the Oracle Enterprise Session Border Controller's address in the Request-URI. The default is disabled. The valid values are:

    - enabled | disabled

For example, you should enable routing to the registrar if your network uses a N Oracle Enterprise Session Border Controller and needs requests to go through its service proxy, which is defined in the registrar host field.

18. teluri-scheme—Enable to convert SIP URIs to tel (resources identified by telephone numbers) URIs.

    If enabled, the requests generated on this SIP interface by the Oracle Enterprise Session Border Controller will have a tel URI scheme instead of the SIP URI scheme. Only the Request, From, and To URIs are changed to the tel scheme. After the dialog is established, the URIs are not changed. The default is disabled. The valid values are:

    - enabled | disabled

19. uri-fqdn-domain—Change the host part of the URIs to the FQDN value set here. If set to enabled, and used with an FQDN domain/host, the requests generated by the Oracle Enterprise Session Border Controller on this SIP interface will have the host part of the URI set to this FQDN value. Only the Request, To, and From URIs are changed. After the dialog is established, the URIs are not changed.
20. trust-mode—Set the trust mode for the SIP interface, which is checked by the Oracle Enterprise Session Border Controller when it receives a message to determine whether the message source is trusted. The default is all. Available options are:

    - all—Trust all SIP elements (sources and destinations) in the realm(s), except untrusted session agents. Untrusted session agents are those that have the trust-me parameter set to disabled.
    - agents-only—Trust only trusted session agents. Trusted session agents are those that have the trust-me parameter set to enabled.
    - realm-prefix—Trust only trusted session agents, and source and destination IP addresses that match the IP interface's realm (or subrealm) address prefix. Only realms with non-zero address prefixes are considered.
    - registered—Trust only trusted session agents and registered endpoints. Registered endpoints are those with an entry in the Oracle Enterprise Session Border Controller's registration cache.
    - none—Trust nothing.

    Session agents must have one or more of the following:

    - global realm
    - same realm as the SIP interface
    - realm that is a subrealm of the SIP interface's realm

21. trans-expire—Set the TTL expiration timer in seconds for SIP transactions. This timer controls the following timers specified in RFC 3261:

    - Timer B—SIP INVITE transaction timeout
    - Timer F—non-INVITE transaction timeout
    - Timer H—Wait time for ACK receipt
    - Timer TEE—Used to transmit final responses before receiving an ACK

    The default is 0. If you leave this parameter set to the default, then the Oracle Enterprise Session Border Controller uses the timer value from the global SIP configuration. The valid range is:

    - Minimum—0
    - Maximum—999999999

22. invite-expire—Set the TTL expiration timer in seconds for a SIP client/server transaction after receiving a provisional response.

    You set this timer for the client and the sever by configuring it on the SIP interface corresponding to the core or access side.

    The default is 0. If you leave this parameter set to the default, then the Oracle Enterprise Session Border Controller uses the timer value from the global SIP configuration. The valid range is:

    - Minimum—0
    - Maximum—999999999

23. max-redirect-contacts—Set the maximum number of contacts or routes for the Oracle Enterprise Session Border Controller to attempt in when it receives a SIP Redirect (3xx Response). The default is 0. If you leave this

parameter set to the default, then the Oracle Enterprise Session Border Controllerwill exercise no restrictions on the number of contacts or routes. The valid range is:

- Minimum—0
- Maximum—10

24. response-map—Enter the name of the SIP response map configuration that you want to apply to this SIP interfaces for outgoing responses. This parameter is blank by default.
25. local-response-map—Enter the name of the SIP response map configuration that you want to apply to this SIP interfaces for locally-generated SIP responses. This parameter is blank by default.
26. options—Optional.

Configuring SIP Ports

To configure SIP ports:

1. From sip-interface, type sip-ports and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-interface)# sip-ports
ACMEPACKET(sip-port)#
```

2. address—Enter the IP address of the host associated with the sip-port entry on which to listen. For example:

```
192.168.11.101
```

3. port—Enter the port number you want to use for this sip-port. The default is 5060. The valid range is:

   - Minimum—1025
   - Maximum—65535

4. transport-protocol—Indicate the transport protocol you want to associate with the SIP port. The default is UDP. The valid values are:

   - TCP—Provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with the retransmission of packets when necessary.
   - UDP—Provides a simple message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
   - TLS—See the Security chapter for more information about configuring TLS.

5. allow-anonymous—Define the allow anonymous criteria for accepting and processing a SIP request from another SIP element.

   The anonymous connection mode criteria includes admission control based on whether an endpoint has successfully registered. Requests from an existing SIP dialog are always accepted and processed. The default is all.

   The following table lists the available options.

   - all—All requests from any SIP element are allowed.
   - agents-only—Only requests from configured session agents are allowed. The session agent must fit one of the following criteria:
     - Have a global realm.
     - Have the same realm as the SIP interface
     - Be a sub-realm of the SIP interface's realm.

       When an agent that is not configured on the system sends an INVITE to a SIP interface, the Oracle Enterprise Session Border Controller:
     - Refuses the connection in the case of TCP.
     - Responds with a 403 Forbidden in the case of UDP.
   - realm-prefix—The source IP address of the request must fall within the realm's address prefix or a SIP interface sub-realm. A sub-realm is a realm that falls within a realm-group tree. The sub-realm is a child (or grandchild, and so on) of the SIP interface realm.

Only realms with non-zero address prefixes are considered. Requests from session agents (as described in the agents-only option) are also allowed.

- registered—Only requests from user agents that have an entry in the registration cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed from any user agent.

  The registration cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the agents-only option) are also allowed.
- register-prefix—Only requests from user agents that have an entry in the Registration Cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed only when the source IP address of the request falls within the realm address-prefix or a SIP interface sub-realm. Only realms with non-zero address prefixes are considered.

  The Registration Cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the agents-only option) are also allowed.

Recurse 305 Only Redirect Action

The Oracle Enterprise Session Border Controller has a SIP feature called redirect action. This is a feature that allows the Oracle Enterprise Session Border Controller, acting as a SIP Proxy or a Session Agent, to redirect SIP messages after receiving a SIP redirect (3xx) response. Previously, for the ACLI objects of sip-interface and session-agent on the Oracle Enterprise Session Border Controller, you could set the redirect-action parameter to proxy or recurse. In Release 6.3 you can additionally set a value of recurse-305-only for the redirect-action parameter.

Redirect Action Process

When the redirect-action parameter is set to proxy, the Oracle Enterprise Session Border Controller sends SIP Redirect responses back to the previous hop (back to the User Agent Client (UAC)) when the User Agent Server (UAS) is not a session agent. The URI in the Contact of the response is changed from the URI that was in the original request.

Note: If the target of the request is a session agent, the session agent's redirect action supercedes that of the SIP interface.

When the redirect-action parameter is set to recurse, if the Oracle Enterprise Session Border Controller receives a SIP redirect (3xx) response on the SIP interface, it automatically redirects all requests to the Contact URI specified in the 3xx response. The responses contain the same Contact URI that was in the original request sent to the UAS.

For example, if UAC X sends an INVITE to the Oracle Enterprise Session Border Controller set up as a SIP proxy, the Oracle Enterprise Session Border Controller forwards the INVITE to UAS Y (Y is not a session agent). Y then responds to the Oracle Enterprise Session Border Controller with a 3xx response (redirection message) with the same URI that was in the original request. This indicates to the Oracle Enterprise Session Border Controller that if it receives any future requests directed toward Y, that it should automatically redirect the request directly to Y. The Oracle Enterprise Session Border Controller then recurses, or repeatedly sends subsequent incoming messages to the Contact URI specified in the Header of the 3xx responses.

When the redirect-action parameter is set to recurse-305-only, if the Oracle Enterprise Session Border Controller receives a 305 SIP redirect response (Use Proxy) on the SIP interface, it automatically redirects all requests to the Contact URI specified in the 305 response. All other 3xx responses are sent back to the previous hop.

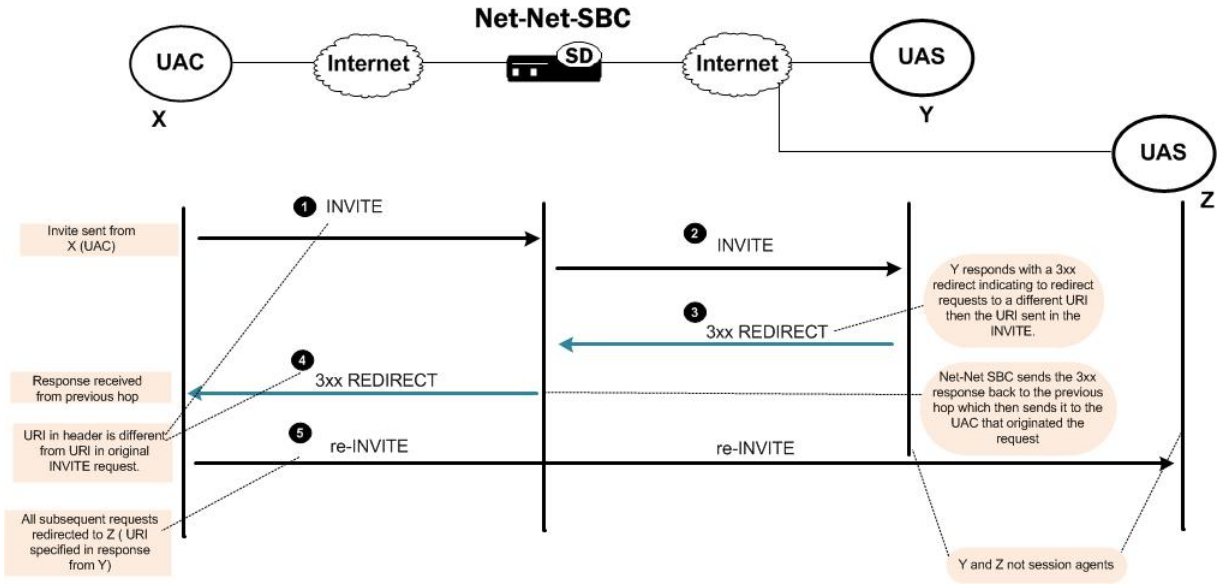
When the UAS is a session agent, the Oracle Enterprise Session Border Controller can send the SIP redirect response back to the UAC using the value in the session agent’s redirect action field. If there are too many UASs to define as individual session agents, or if the UASs are Hosted NAT Traversal (HNT) endpoints, and SIP redirect responses need to be proxied for UASs that are not session agents, you can set the behavior at the SIP interface level.

Redirect-Action Set to Proxy

The following occurs if you set the redirect-action parameter to proxy on the Oracle Enterprise Session Border Controller:

1. X (UAC) sends an INVITE to the Oracle Enterprise Session Border Controller.
2. The Oracle Enterprise Session Border Controller forwards the INVITE to Y (UAS).
3. Y sends the 3xx REDIRECT response to the Oracle Enterprise Session Border Controller with a different URI in the message header.
4. The Oracle Enterprise Session Border Controller forwards the 3xx REDIRECT response to the previous hop. X receives the 3xx REDIRECT response from the previous hop.
5. X redirects all subsequent requests to the URI in the message header received from Y.

The following illustration shows an example of a dialog between X, Y, Z, and the Oracle Enterprise Session Border Controller during a redirect-action session set to proxy.

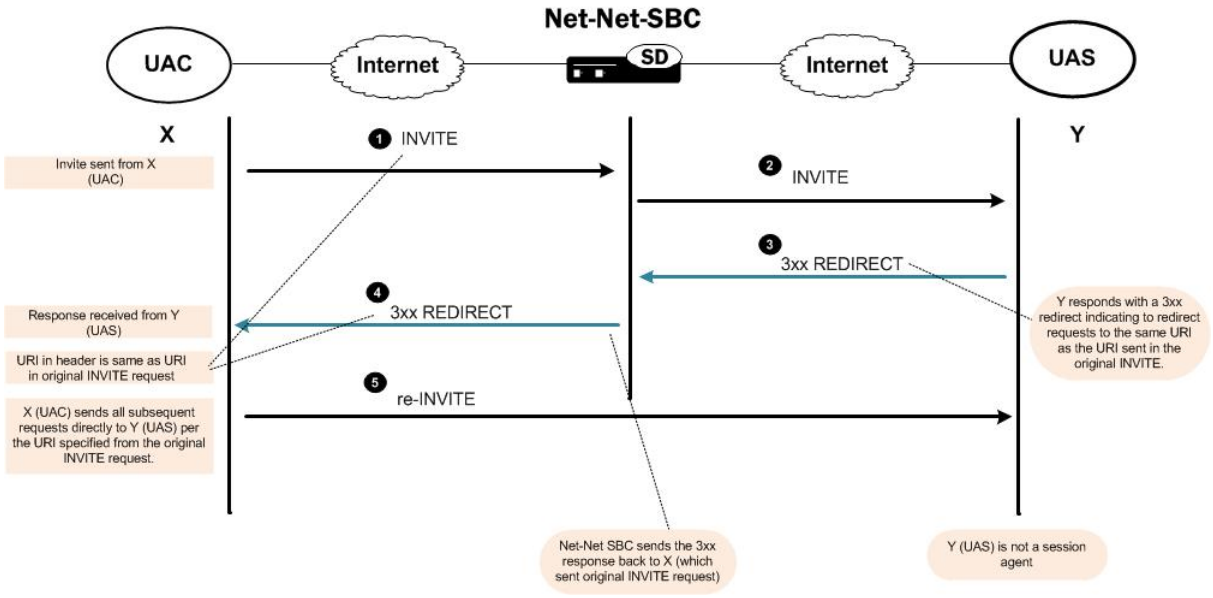


Redirect-Action Set to Recurse

The following occurs if you set the redirect-action parameter to recurse on the Oracle Enterprise Session Border Controller:

1. X (UAC) sends an INVITE to the Oracle Enterprise Session Border Controller.
2. The Oracle Enterprise Session Border Controller forwards the INVITE to Y (UAS).
3. Y sends the 3xx REDIRECT response to the Oracle Enterprise Session Border Controller with the same URI as the URI sent in the original request.
4. The Oracle Enterprise Session Border Controller forwards the 3xx REDIRECT response to X (UAC).
5. X (UAC) sends all subsequent requests directly to Y (UAS) per the URI specified from the original INVITE request.

The following illustration shows an example of a dialog between X, Y, and the Oracle Enterprise Session Border Controller during a redirect-action session set to recurse.

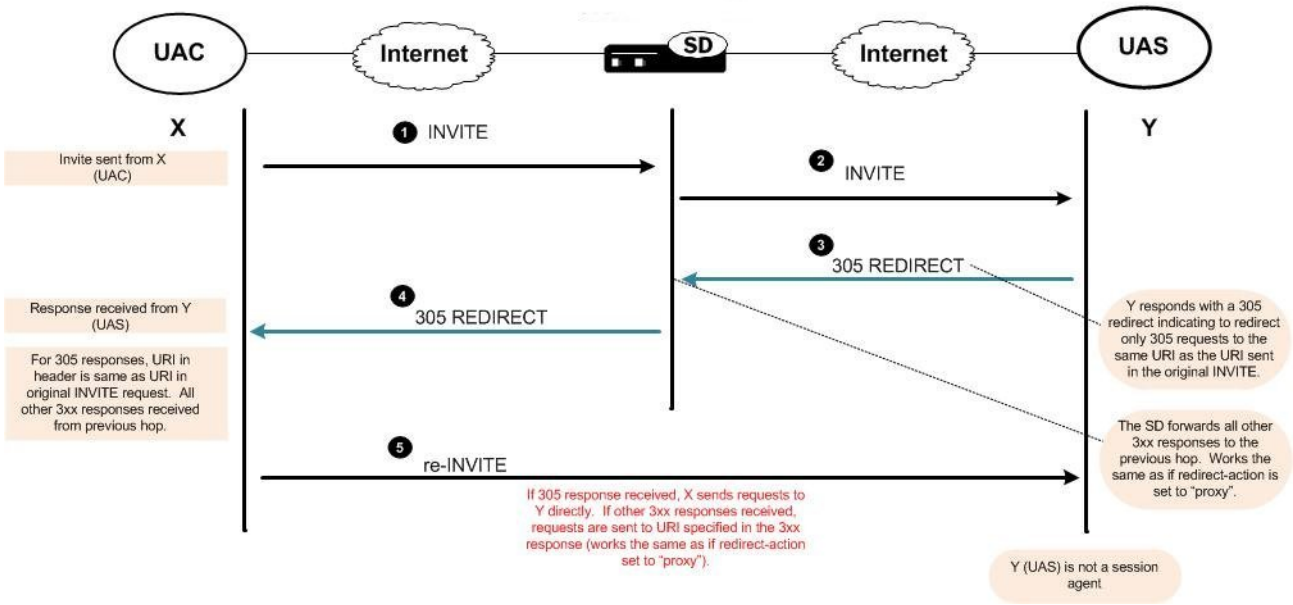


Redirect-Action Set to Recurse-305-Only

The following occurs if you set the redirect-action parameter to recurse-305-only on the Oracle Enterprise Session Border Controller:

1. X (UAC) sends an INVITE to the Oracle Enterprise Session Border Controller.
2. The Oracle Enterprise Session Border Controller forwards the INVITE to Y (UAS).
3. Y sends a 305 REDIRECT response to the Oracle Enterprise Session Border Controller with the same URI as the URI sent in the original request.
4. The Oracle Enterprise Session Border Controller forwards the 305 REDIRECT response to X (UAC).
5. If 305 response received, X sends requests to Y directly. If other 3xx responses received, requests are sent to URI specified in the 3xx response (works the same as if redirect-action set to proxy).

   The following illustration shows an example of a dialog between X, Y, and the Oracle Enterprise Session Border Controller during a redirect-action session set to recurse-305-only.



Redirect Configuration for SIP Interface

You can configure the Oracle Enterprise Session Border Controller to redirect requests from a UAC to a UAS using the URI in 305 responses only. You can use the ACLI at the paths **session-router** > **sip-interface or session-router** > **session-agent.**

To configure the redirect-action feature on the SIP interface on the Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter to access the SIP interface-related configurations. The system prompt changes to let you know that you can begin configuring individual parameters for this object.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Enter redirect-action followed by the following value:

   - recurse-305-only

```
ACMEPACKET(sip-interface)# redirect-action recurse-305-only
```

   When the Oracle Enterprise Session Border Controller receives a 305 SIP redirect response (Use Proxy) on the SIP interface, it automatically redirects all requests to the Contact URI specified in the 305 response. All 3xx responses other than 305 responses are sent back to the previous hop.

   To disable this feature, enter redirect-action and press Enter without entering a value.

```
ACMEPACKET(sip-interface)# redirect-action
```

Embedded Routes in Redirect Responses

When the Oracle Enterprise Session Border Controller recurses as the result of a redirect (3xx) response, the server might need to specify one or more intermediate hops. These hops are reflected in the Contact header for the 3xx response using embedded route headers and look like this:

```
Contact: <sip:touser@server.example.com?Route=%3Cproxy.example.com%Blr%3E>
```

The Contact header shows that the request should be sent to server.example.com using proxy.example.com.

You can configure your Oracle Enterprise Session Border Controller to specify that embedded headers in 3xx Contact headers are to be included in new requests such that they are tied to a session agent representing the new target (server.example.com). This behavior requires you to set the request-uri-headers parameter.

However, you can also use the use-redirect-route in global SIP configuration's options parameter so that the embedded Route header is used as the next hop to receive the new request.

When you configure this new option, the Oracle Enterprise Session Border Controller constructs a new request using the redirect Contact, and the SIP URI from the Contact becomes the Request-URI. Then, the system inserts the embedded routes as Route headers in the, using the same order in which they appeared in the redirect Contact. Afterward, the Oracle Enterprise Session Border Controller determines the next hop in the same way it does with any other request. If the first route is a loose route (i.e., it has the lr URI parameter), then the Oracle Enterprise Session Border Controller sends a request to host indicated in the first route. Otherwise, strict routing applies, and the Oracle Enterprise Session Border Controller sends the request to the host indicated in the Request-URI.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration that you want to edit.

4. options—Set the options parameter by typing options, a Space, and then the option name.

```
ACMEPACKET(sip-config)# options use-redirect-route
```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

SIP PRACK Interworking

When you configure your Oracle Enterprise Session Border Controller with PRACK interworking for SIP, you enable it to interwork between endpoints that support RFC 3262, *Reliability of Provisional Responses in the Session Initiation Protocol*, and those that do not.

As its title indicates, RFC 3262 defines a reliable provisional response extension for SIP INVITEs, which is the 100rel extension tag. While some endpoints do not support the RFC, other SIP implementations require compliance with it. A session setup between two such endpoints fails. However, you can configure your Oracle Enterprise Session Border Controller to supply the provisional response on behalf of endpoints that do not support it—and thereby enable sessions between those endpoints and the ones requiring RFC 3262 compliance.

You need to configure PRACK interworking for a SIP interface associated with the endpoints that need RFC 3262 support. To enabled the feature, you set the 100rel-interworking option. The Oracle Enterprise Session Border Controller applies PRACK interworking for either the UAC or the UAS. TheOracle Enterprise Session Border Controller checks to see whether or not it needs to apply PRACK interworking when an INVITE arrives at the ingress or egress SIP interface with the option enabled. First, it checks the Require header for the 100rel tag; if not found there, it checks the Supported header.

Since there is a slight difference in the application of this feature between the UAC and UAS, this section explains both.

Note: If SDP is included in a PRACK request sent to a SIP interface where PRACK interworking is enabled, it will not be responded to, nor will any SDP be included in the locally-generated 200 OK to that PRACK.

UAC-Side PRACK Interworking

The Oracle Enterprise Session Border Controller applies PRACK interworking on the UAC side when:

- A SIP INVITE does not contain a 100rel tag in a Require or Supported header
- The ingress SIP interface is enabled with the 100rel-interworking option
- The UAS fails to send reliable provisional responses

When it is to forward a non-reliable response to a UAC that requires RFC 3262 support, the Oracle Enterprise Session Border Controller converts the non-reliable response to a reliable one by adding the 100rel tag to the Require header and adding an Rseq header to the response. Further, the Oracle Enterprise Session Border Controller adds a Require header (complete with the100rel tag) if there is not one already in the response, and then also adds Rseq header.

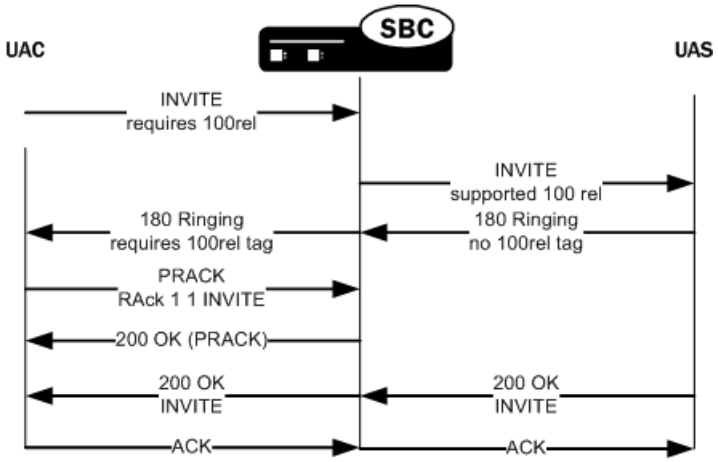
Note that the Oracle Enterprise Session Border Controller sets the value of the Rseq header as 1 for the first provisional response, and then increments it by 1 for each subsequent provisional response. It also adds the PRACK method to the Allow header when that header appears.

The Oracle Enterprise Session Border Controller retransmits the converted reliable provisional response in accordance with RFC 3262, until it receives a PRACK request. For the initial timeout for retransmission, the Oracle Enterprise Session Border Controller uses the value you set in the init-timer parameter in the global SIP

configuration. It stops retransmitting when either it receives a transmission, or when the ingress SIP interface's trans-expire timer elapses.

If it never receives a PRACK, the Oracle Enterprise Session Border Controller does not generate an error response to the INVITE, relying instead on the downstream UAS to produce a final response.

The call flow for this application looks like this:



UAS-Side PRACK Interworking

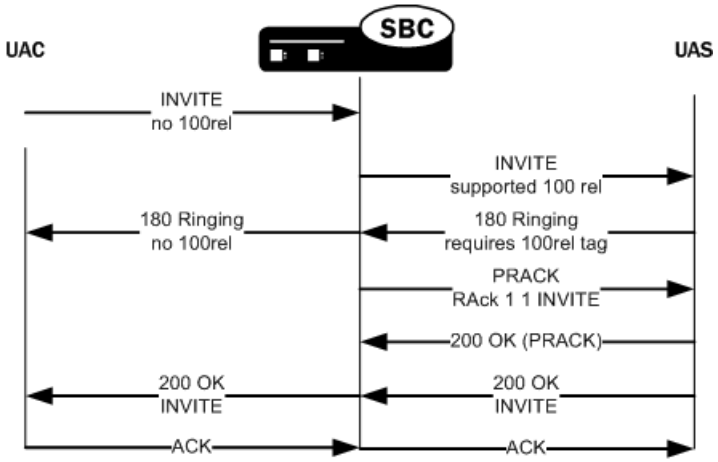
The Oracle Enterprise Session Border Controller applies PRACK interworking on the UAS side when:

- A SIP INVITE contains the 100rel tag in a Require or Supported header
- The egress SIP interface is enabled with the 100rel-interworking option
- The UAS does send reliable provisional responses

When the UAC does not support RFC 3262, the Oracle Enterprise Session Border Controller generates a PRACK request to acknowledge the response. It also converts the response to non-reliable by removing the 100 rel tag from the Require header and removing the RSeq header from the response.

In the case of the UAS, the Oracle Enterprise Session Border Controller matches the PRACK to a converted reliable provisional response using the PRACK's RAck header. If it finds a matching response, the Oracle Enterprise Session Border Controller generates a 200 OK to the PRACK. And if it finds no match, then it generates a 481 Call Leg/ Transaction Does Not Exist response. The Oracle Enterprise Session Border Controller generates a 400 Bad Request response if either the RAck is not in the PRACK request or it is not formatted properly.

The call flow for this application looks like this:



PRACK Interworking Configuration

You enable PRACK interworking for ingress and egress SIP interfaces. Be sure you know on what side, ingress or egress, you need this feature applied.

To configure PRACK interworking for a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. If you are editing an existing configuration, select the one on which you want to enable this feature.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. options—Set the options parameter by typing options, a Space, the option name 100rel-interworking with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-interface)# options +100rel-interworking
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Global SIP Timers

This section explains how to configure SIP retransmission and expiration timers.

Note: you can also set timers and counters per SIP interface.

Overview

SIP timers define the transaction expiration timers, retransmission intervals when UDP is used as a transport, and the lifetime of dynamic TCP connections. The retransmission and expiration timers correspond to the timers defined in RFC 3261.

- init timer: is the initial request retransmission interval. It corresponds to Timer T1 in RFC 3261.

  This timer is used when sending requests over UDP. If the response is not received within this interval, the request is retransmitted. The retransmission interval is doubled after each retransmission.
- max timer: is the maximum retransmission interval for non-INVITE requests. It corresponds to Timer T2 in RFC 3261.

  The retransmission interval is doubled after each retransmission. If the resulting retransmission interval exceeds the max timer, it is set to the max timer value.
- trans expire: is the transaction expiration timer. This value is used for timers B, D, F, H and J as defined in RFC 3261.
- invite expire: defines the transaction expiration time for an INVITE transaction after a provisional response has been received. This corresponds to timer C in RFC 3261.

  If a final response is not received within this time, the INVITE is cancelled. In accordance with RFC 3261, the timer is reset to the invite expire value when any additional provisional responses are received.
- Inactive dynamic conn timer defines the idle time of a dynamic TCP connection before the connection is torn down. Idle is defined as not transporting any traffic. There is no timer in RFC 3261 corresponding to this function.

Timers Configuration

To configure timers:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. init-timer—Enter the initial timeout value in milliseconds for a response to an INVITE request, and it applies to any SIP request in UDP. In RFC 3261, this value is also referred to as TIMER_T1. The default is 500. The valid range is:

   - Minimum—0
   - Maximum—999999999

5. max-timer—Enter the maximum transmission timeout (T2) for SIP in milliseconds.

   When sending SIP over UDP, a re-transmission timer is used. If the timer expires and the message is re-transmitted, the re-transmission timer is then set to twice the previous value (but will not exceed the maximum timer value). Using the default values of 500 milliseconds and 4000 milliseconds, the re-transmission timer is 0.5, then 1, 2, and finally 4. The incrementing continues until the transmission expire timer activates. The default is 4000. The valid range is:

   - Minimum—0
   - Maximum—999999999

6. trans-expire—Enter the transaction expire timeout value (Timer B) in seconds to set the time for SIP transactions to live. The same value is used for Timers D, F, H and J. The default is 32. The valid range is:

   - Minimum—0
   - Maximum—999999999

7. invite-expire—Enter the invite expire timeout value (Timer C) in seconds to indicate the time for SIP client transaction will live after receiving a provisional response. The default is 180. The valid range is:

   - Minimum—0
   - Maximum—999999999

8. inactive-dynamic-conn—Enter the inactive dynamic connection value in seconds to set the time limit for inactive dynamic connections.

   If the connection between the SIP proxy and a session agent is dynamic (for example, through dTCP), and the connection has been idle for the amount of time specified here, the SIP proxy breaks the connection. Idle is defined as not transporting any traffic. The default value is 32. The valid range is:

   - Minimum—0
   - Maximum—999999999

   Note: Setting this parameter to 0 disables this parameter.

   The following example shows SIP config timer values for a peering network. Some parameters are omitted for brevity.

```
sip-config
        state                           enabled
        operation-mode                  dialog
dialog-transparency        disabled
home-realm-id                   acme
```

```
        egress-realm-id
        nat-mode                        Public
        registrar-domain
        registrar-host
        registrar-port                  0
        init-timer                      500
        max-timer                       4000
        trans-expire                    32
        invite-expire                   180
        inactive-dynamic-conn           32
```

SIP Timers Discreet Configuration

Previous releases controlled various SIP timers with a single ACLI command, trans-expire, available in both sip-config and sip-interface modes. When executed in sip-config mode, the command essentially established a global default transaction expiration timer value. Executed at the sip-interface level, the command established a local, interface-specific value that overrode the global default.

Specific timers controlled by trans-expire are as follows:

Timer B, the INVITE transaction timeout timer, defined in Section 17.1.1.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

Timer D, the Wait-Time for response retransmitals timer, defined in Section 17.1.1.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

Timer F, the non-INVITE transaction timeout timer, defined in Section 17.1.2.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

Timer H, the Wait-Time for ACK receipt timer, defined in Section 17.2.1 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

Timer J, the Wait-Time for non-INVITE requests timer, defined in Section 17.2.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

A new ACLI command (initial-inv-trans-expire) that enables user control over SIP Timer B for initial INVITE transactions. Other timers, namely B for non-initial INVITEs, D, F, H, and J remain under the control of trans-expire.

Use initial-inv-trans-expire in the sip-config configuration mode, to establish a global, default transaction timeout value (expressed in seconds) used exclusively for initial INVITE transactions.

```
ACMEPACKET(sip-config)# initial-inv-trans-expire 4
ACMEPACKET(sip-config)#
```

Allowable values are integers within the range 0 (the default) through 999999999. The default value, 0, indicates that a dedicated INVITE Timer B is not enabled. Non-default integer values enable a dedicated Timer B and set the timer value.

The default value retains compatibility with previous operational behavior in that Timers B, D, F, H, and J all remain subject to the single timer value set by trans-expire. However, when initial-inv-trans-expire is et to a supported non-zero value, SIP Timer B as it applies to initial INVITEs, assumes that value rather than the value assigned by trans-expire. This functionality is available in both sip-config and in sip-interface objects.

If a dedicated Timer B is enabled at the sip-config level, you can use initial-inv-trans-expire in the sip-interface configuration mode, to establish a local interface-specific Timer B timeout value that overrides the global default value.

```
ACMEPACKET(sip-interface)# initial-inv-trans-expire 8
ACMEPACKET(sip-interface)#
```

Session Timer Support

The Oracle Enterprise Session Border Controller partially supports RFC4028 by establishing session timers without participating in the session timer negotiation.

When a 2xx response to a Session Refresh Request arrives, the Oracle Enterprise Session Border Controller will start a new timer or refresh the existing timer using the value of the Session-Expires header. When the session timer expires, the Oracle Enterprise Session Border Controller will send a BYE to both the upstream and downstream endpoints.

When accounting is configured, the Oracle Enterprise Session Border Controller will also send a RADIUS STOP record with Acct-Terminate-Cause=Session-Timeout.

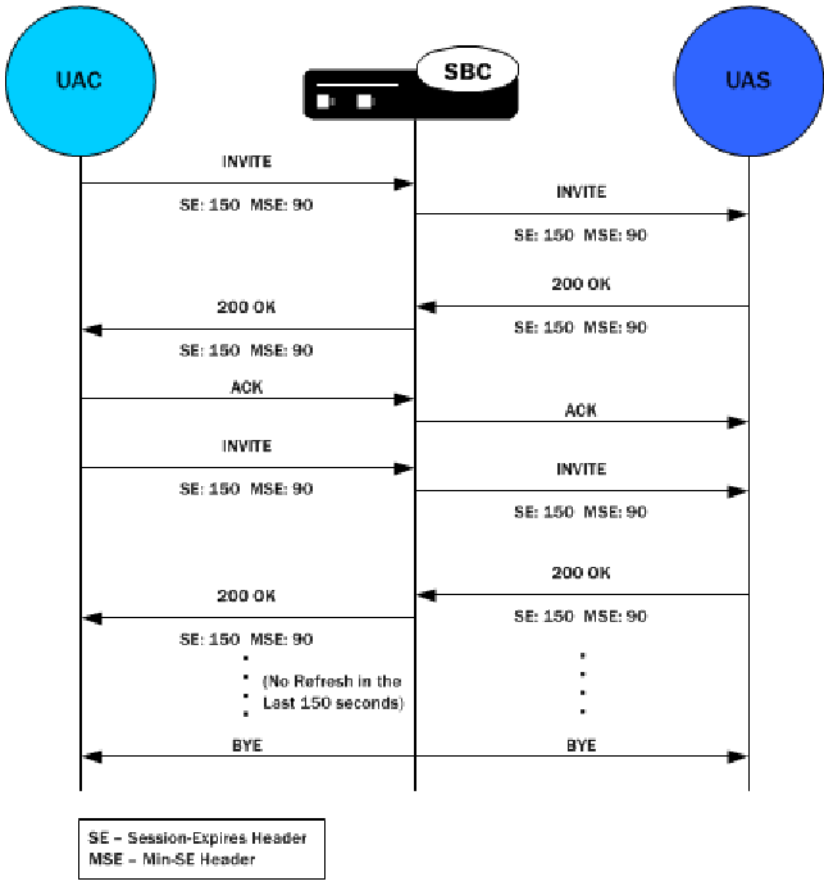
Call Flow Example

The UAS obtains the value from the Session-Expires header field in a 2xx response to a session refresh request that it sends.

Proxies and UACs obtain this value from the Session-Expires header field in a 2xx response to a session refresh request that they receive.

Once the session timer runs out, the Oracle Enterprise Session Border Controller sends a BYE to both the UAC and the UAS to clear the session.

Enable this feature by adding the session-timer-support option to the sip config.



SIP Per-User CAC

The Oracle Enterprise Session Border Controller's call admission control (CAC) supports an enhanced degree of granularity for SIP sessions.

Without this feature enabled, the Oracle Enterprise Session Border Controller performs call admission control (CAC) based on:

- Bandwidth limits configured in realms and nested realms
- Number of media flows available through the steering pool per realm
- Number of inbound sessions configured for a SIP session agent
- Number of total sessions (inbound and outbound) per SIP session agent
- Use of the Oracle Enterprise Session Border Controller's support for common open policy service (COPS), allowing the Oracle Enterprise Session Border Controller to perform CAC based on the policies hosted in an external policy server

These methods provide a basic level of call admission control in order to ensure that a SIP session agent's capacity is not exceeded. You can also ensure that signaling and media bandwidth capacities are not exceeded for physical trunks and peers.

With this feature enabled, the Oracle Enterprise Session Border Controller changes behavior so that it will only allow the configured number of calls or total bandwidth to and from each user in a particular realm. The overall realm bandwidth and steering pool limits still apply, and as before, the Oracle Enterprise Session Border Controller still rejects users who might be within their CAC limitations if accepting them with exceed the bandwidth limitations for parent or child realms and steering pools.

For SIP sessions, the Oracle Enterprise Session Border Controller now keeps track of the amount of bandwidth a user consumes and the number of active sessions per address of record (AoR) or per IP address, depending on the CAC mode you select (either aor or ip). When an endpoint registers with the Oracle Enterprise Session Border Controller, theOracle Enterprise Session Border Controller allots it a total amount of bandwidth and total number of sessions.

This section describes the details of how SIP per user CAC works.

You should note that the functionality this section describes only works if you enable registration caching on your Oracle Enterprise Session Border Controller.

For SIP sessions, the Oracle Enterprise Session Border Controller now keeps track of the amount of bandwidth a user consumes and the number of active sessions per address of record (AoR) or per IP address, depending on the CAC mode you select (either aor or ip). When an endpoint registers with the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller allots it a total amount of bandwidth and total number of sessions.

Per User CAC Modes

There are three modes that you can set for this feature, and each has an impact on how the other two per-user-CAC parameters are implemented:

- none—No per user CAC is performed for users in the realm.
- aor—The Oracle Enterprise Session Border Controller performs per user CAC according to the AoR and the contact associated with that AoR for users in the realm.
- ip—The Oracle Enterprise Session Border Controller performs per user CAC according to the IP address and all endpoints that are sending REGISTER messages from the IP address for users in the realm.

Per User CAC Sessions

You can set the number of CAC for sessions per user in the realm configuration. Depending on the CAC mode you set, the sessions are shared between contacts for the same AoR or the endpoints behind the same IP address.

When it receives an INVITE, the Oracle Enterprise Session Border Controller determines the registration entry for the calling endpoint and the registration for the called endpoint. It then decides if session can be established between the

two. If it can, the Oracle Enterprise Session Border Controller establishes the session and changes the active session count for the calling and called endpoints. The count is returned to its original value once the session is terminated.

Per User CAC Bandwidth

You can set the per user CAC bandwidth in realm configuration, too, and it is handled much the same way that the sessions are handled. That is, depending on the CAC mode you set, the bandwidth is shared between contacts for the AoR or the endpoints behind the same IP address. All endpoints must be registered with the Oracle Enterprise Session Border Controller.

When it receives a Request with SDP, the Oracle Enterprise Session Border Controller checks to see if there is enough bandwidth for the calling endpoint and for the called endpoint. The Oracle Enterprise Session Border Controller assumes that the bandwidth usage is symmetric, and it uses the maximum bandwidth configured for the codec that it finds in the Request. In the event that there are multiple streams, the Oracle Enterprise Session Border Controller determines the total bandwidth required for all of the streams. If the required bandwidth exceeds what is available for either endpoint, the Oracle Enterprise Session Border Controller rejects the call (with a 503 error response). If the amount of available bandwidth is sufficient, then the used bandwidth value is increased for both the registered endpoints: calling and called. Any mid-session requests for changes in bandwidth, such as those caused by modifications in codec use, are handled the same way.

The Oracle Enterprise Session Border Controller also keeps track of the bandwidth usage on a global level. When the call terminates, the bandwidth it was consuming is returned to the pool of available bandwidth.

Notes on HA Nodes

This feature has been implemented so that a newly active system is able to perform SIP per user CAC. The standby Oracle Enterprise Session Border Controller is updated with the appropriate parameters as part of the SIP session update.

SIP per User CAC Configuration

Note that you must enable registration caching for this feature to work.

To configure SIP per user CAC:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to want to add SIP per user CAC.

```
ACMEPACKET(realm-config)# select
```

5. user-cac-mode—Set this parameter to the per user CAC mode that you want yo use. The default value is none. The valid values are:
   - none—No user CAC for users in this realm
   - aor—User CAC per AOR
   - ip—User CAC per IP
6. user-cac-sessions—Enter the maximum number of sessions per user for dynamic flows to and from the user. The default is 0. Leaving this parameter set to its means that there is unlimited sessions, meaning that the per user CAC feature is disabled in terms of the constraint on sessions. The valid range is:
7. Minimum—0

8. Maximum—999999999
9. user-cac-bandwidth—Enter the maximum bandwidth per user for dynamic flows to and from the user. The default is 0 and leaving this parameter set to the default means that there is unlimited bandwidth, meaning that the per user CAC feature is disabled in terms of the constraint on bandwidth. The valid range is:
   - Minimum—0
   - Maximum—999999999

SIP Per-Realm CAC

Building on the Oracle Enterprise Session Border Controller's pre-existing call admission control methods, CAC can be performed based on how many minutes are being used by SIP or H.323 calls per-realm for a calendar month.

In the realm configuration, you can now set a value representing the maximum number of minutes to use for SIP and H.323 session using that realm. Although the value you configure is in minutes, the Oracle Enterprise Session Border Controller performs CAC based on this value to the second. When you use this feature for configurations with nested realms, the parent realm will have the total minutes for all its child realms (i.e., at least the sum of minutes configured for the child realms).

The Oracle Enterprise Session Border Controller calculates the number of minutes used when a call completes, and counts both call legs for a call that uses the same realm for ingress and egress. The total time attributed to a call is the amount of time between connection (SIP 200 OK) and disconnect (SIP BYE), regardless of whether media is released or not; there is no pause for calls being placed on hold.

If the number of minutes is exhausted, the Oracle Enterprise Session Border Controller rejects calls with a SIP 503 Service Unavailable message (including additional information "monthly minutes exceeded). In the event that the limit is reached mid-call, the Oracle Enterprise Session Border Controller continues with the call that pushed the realm over its threshold but does not accept new calls. When the limit is exceeded, the Oracle Enterprise Session Border Controller issues an alarm and sends out a trap including the name of the realm; a trap is also sent when the alarm condition clears.

Note: The Oracle Enterprise Session Border Controller does not reject GETS/NSEP calls based on monthly minutes CAC.

You can change the value for minutes-based CAC in a realm configuration at any time, though revising the value downward might cause limits to be reached. This value resets to zero (0) at the beginning of every month, and is checkpointed across both system in an HA node. Because this data changes so rapidly, however, the value will not persist across and HA node if both systems undergo simultaneous failure or reboot.

You can use the ACLI show monthly minutes <realm-id> command (where <realm-id> is the realm identifier of the specific realm for which you want data) to see how many minutes are configured for a realm, how many of those are still available, and how many calls have been rejected due to exceeding the limit.

SIP per Realm CAC Configuration

This section shows you how to configure minutes-based CAC for realms and how to display minutes-based CAC data for a specific realm.

Enabling Realm-Based CAC

Note that setting the new monthly-minutes parameters to zero (0), or leaving it set to its default of 0, disables this feature.

To configure minutes-based CAC:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to want to add SIP per user CAC.

```
ACMEPACKET(realm-config)# select
```

5. monthly-minutes—Enter the number of minutes allowed during a calendar month in this realm for SIP and H.323 calls. By default, this parameter is set to zero (0), which disabled monthly minutes-based CAC. You can enter a value as high as 71582788.
6. Save and activate your configuration.

Viewing Realm-Based CAC Data

Use the ACLI show monthly-minutes command to see the following information:

- How many minutes are configured for a realm
- How many of those are still available
- How many calls have been rejected due to exceeding the limit

To view information about SIP per user CAC using the IP address mode:

In either User or Superuser mode, type show monthly-minutes <realm-id>, a Space, and the IP address for which you want to view data. Then press Enter. The <realm-id> is the realm identifier for. the realm identifier of the specific realm for which you want data

```
ACMEPACKET# show monthly-minutes private_realm
```

SIP Options Tag Handling

This section explains how to configure SIP options on a global or per-realm level and how to specify whether the feature treatment applies to traffic inbound to or outbound from a realm, or both.

SIP extensions that require specific behavior by UAs or proxies are identified by option tags. Option tags are unique identifiers used to designate new options (for example, extensions) in SIP. These option tags appear in the Require, Proxy-Require, and Supported headers of SIP messages.

Option tags are compatibility mechanisms for extensions and are used in header fields such as Require, Supported, Proxy-Require, and Unsupported in support of SIP.

The option tag itself is a string that is associated with a particular SIP option (i.e., an extension). It identifies this option to SIP endpoints.

Overview

The SIP specification (RFC 3261) requires that the Oracle Enterprise Session Border Controller B2BUA reject any request that contains a Require header with an option tag the Oracle Enterprise Session Border Controller does not support. However, many of these extensions operate transparently through the Oracle Enterprise Session Border Controller's B2BUA. You can configure how SIP defines the Oracle Enterprise Session Border Controllers B2BUA treatment of specific option tags.

Also, there might be certain extensions that an endpoint indicates support for by including the option tag in a Supported header. If you do not want a given extension used in your network, the you can configure SIP option tag handling to remove the undesired option tag from the Supported header. You can also specify how option tags in Proxy-Require headers are to be treated.

Configuration Overview

You configure the SIP feature element to define option tag names and their treatment by the Oracle Enterprise Session Border Controller when the option tag appears in a Supported header, a Require header, and a Proxy-Require header. If an option tag is encountered that is not configured as a SIP feature, the default treatments apply. You only need to configure option tag handling in the SIP feature element when non-default treatment is required.

You can specify whether a SIP feature should be applied to a specific realm or globally across realms. You can also specify the treatment for an option based on whether it appears in an inbound or outbound packet. Inbound packets are those that are coming from a realm to the Oracle Enterprise Session Border Controller and outbound packets are those which are going from the Oracle Enterprise Session Border Controller to the realm.

The following tables lists the SIP option tag parameters you need to configure.

Parameter	Description
name	SIP feature tag name
realm	Realm name with which the feature will be associated. To make the feature global, leave the field empty.
support mode inbound	Action for tag in Supported header in an inbound packet.
require mode inbound	Action for tag in Require header in an inbound packet
proxy require mode inbound	Action for tag in Proxy-Require header in an inbound packet
support mode outbound	Action for tag in Supported header in an outbound packet
require mode outbound	Action for tag in Require header in an outbound packet
proxy require mode outbound	Action for tag in Proxy-Require header in an outbound packet

SIP Option Tag Handling Configuration

To configure SIP option tag handling:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-feature and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-feature
   ACMEPACKET(sip-feature)#
   ```

   From this point, you can configure SIP option tags parameters. To view all sip-feature parameters, enter a ? at the system prompt.

4. name—Enter a name for the option tag that will appear in the Require, Supported, or Proxy-Require headers of inbound and outbound SIP messages.

   You must enter a unique value.

   Note: Valid option tags are registered with the IANA Protocol Number Assignment Services under Session Initiation Protocol Parameters. Because option tags are not registered until the SIP extension is published as a RFC, there might be implementations based on Internet-Drafts or proprietary implementations that use unregistered option tags.

5. realm—Enter the name of the realm with which this option tag will be associated. If you want to apply it globally across realms, leave this parameter blank.
6. support-mode-inbound—Optional. Indicate the support mode to define how the option tag is treated when encountered in an inbound SIP message's Supported header. The default value is pass. Valid values are:

- pass—Indicates the B2BUA should include the tag in the corresponding outgoing message.
- strip—Indicates the tag should not be included in the outgoing message. Use strip if you do not want the extension used.

7. require-mode-inbound—Optional. Indicate the require mode to define how the option tag is treated when it is encountered in an inbound SIP message's Require header. The default value is reject. The valid values are:

   - pass—Indicates the B2BUA should include the tag in the corresponding outgoing message.
   - reject—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

8. require-mode-inbound—Optional. Indicate the require proxy mode to define how the option tag is treated when encountered in an incoming SIP message's Proxy-Require header. The default is reject. The valid values are:

   - pass—Indicates the B2BUA should include the tag in the corresponding outgoing message.
   - reject—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

9. support-mode-outbound—Optional. Indicate the support mode to define how the option tag is treated when encountered in an outbound SIP message's Supported header. The default value is pass. Valid values are:

   - pass—Indicates the B2BUA should include the tag.
   - strip—Indicates the tag should not be included in the outgoing message. Use strip if you do not want the extension used.

10. require-mode-outbound—Optional. Indicate the require mode to define how the option tag is treated when it is encountered in an outbound SIP message's Require header. The default value is reject. Valid values are:

    - pass—Indicates the B2BUA should include the tag.
    - reject—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

11. require-mode-outbound—Optional. Indicate the require proxy mode to define how the option tag is treated when encountered in an outgoing SIP message's Proxy-Require header. The default value is reject. The valid values are:

    - pass—Indicates the B2BUA should include the tag.
    - reject—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

    The following example shows SIP option tag handling configured for non-default treatment of option tags.

```
sip-feature
        name                            newfeature
        realm                           peer-1
        support-mode-inbound            Strip
        require-mode-inbound            Reject
        proxy-require-mode-inbound      Pass
        support-mode-outbound           Pass
        require-mode-outbound           Reject
        proxy-require-mode-outbound     Reject
        last-modified-date              2004-12-08 03:55:05
```

Replaces Header Support

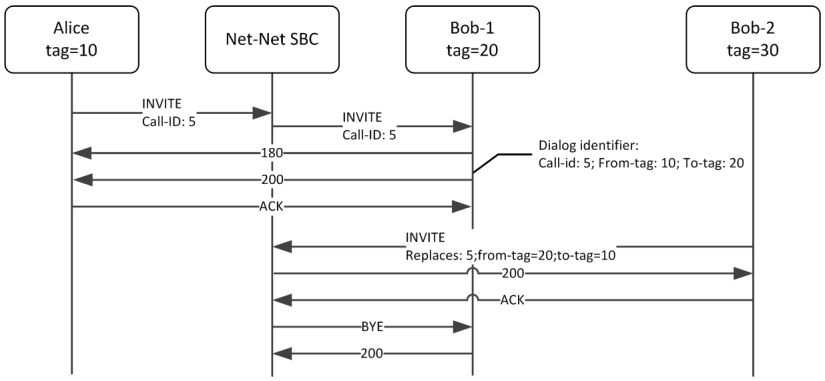
The Oracle Enterprise Session Border Controller supports the Replaces: header in SIP messages according to RFC 3891. The header, included within SIP INVITE messages, provides a mechanism to replace an existing early or established dialog with a different dialog which can be used for services such as call parking, attended call transfer and various conferencing features.

The Replaces: header indicates the dialog it wishes to replace by containing the corresponding dialog identifier. The identifier includes the triplet of the from tag, to tag, and call id. The orientation of endpoint-created tags, as from-tag and to-tag will match each of the two dialogs for a standard call. Thus the Replaces: header from an endpoint that

indicates it wants to assume the dialog between the Oracle Enterprise Session Border Controller and Bob-1 appears as:

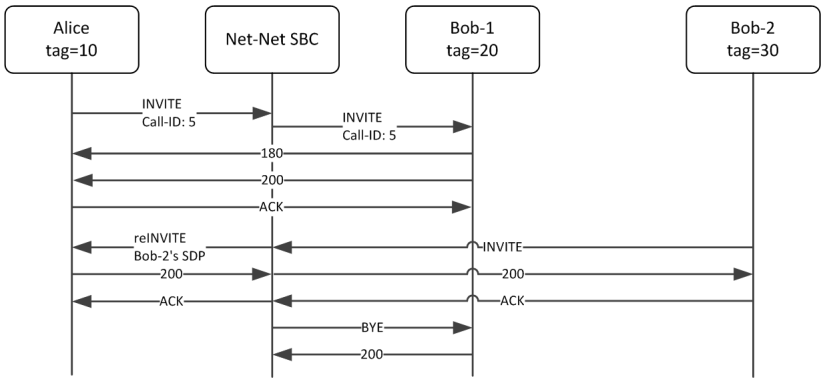
```
Replaces:5555;from-tag=20;to-tag=10
```

The Oracle Enterprise Session Border Controller validates that dialog identifier by matching an existing dialog and tries to install the new endpoint and remove the old endpoint by gracefully ending that dialog with a BYE. The replaces INVITE must come from an edpoint in the same realm as the endpoint it is replacing. If the UA sending the Replaces header is in a different realm as the original call leg (or indicates such architecture via a malformed Replaces: header), the Oracle Enterprise Session Border Controller replies to the Replaces: endpoint with a 481 Missing Dialog. Refer to the following diagram for the standard case.



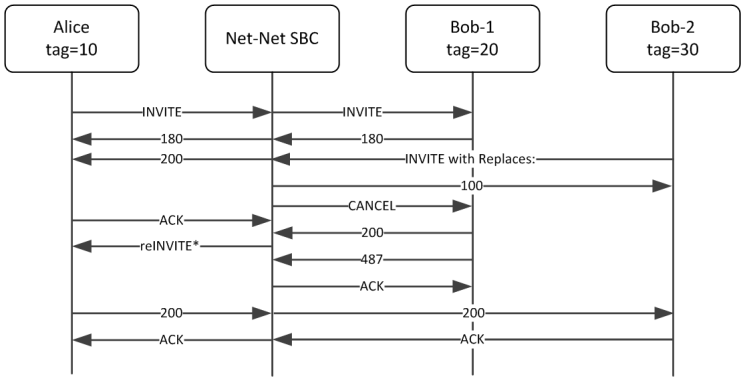
New SDP Parameters in INVITE with Replaces

When an INVITE with Replaces: header is received, the media parameters in the new SDP are compared against the SDP of the dialog to be replaced. If any portion of the SDPs (excluding the session-origin line) is different, then the media must be renegotiated. The Oracle Enterprise Session Border Controller sends a re-INVITE with the new SDP to the dialog opposite of the one being replaced as shown below. If the re-INVITE fails for any reason, then the original dialogs will remain.



Early Dialog Replacement

An INVITE with Replaces: header can replace an early dialog. That is, a dialog where the final 2xx class response to INVITE request has not arrived yet. The Oracle Enterprise Session Border Controller completes the originating side of the call with a 200 OK. The original dialog with the terminator is cancelled. SDP from the new terminator can be renegotiated if it changes.



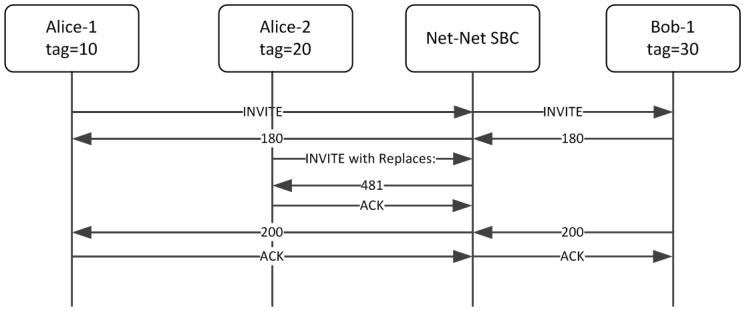
The SDP from the original 183 response is used for the 200 response back to the originator if present to complete the early transaction. If reliable provisional messages are used, then no SDP is included in the 200 response.

If no SDP is present in any of the provisional messages, then the Oracle Enterprise Session Border Controller constructs it from the original offer and modifying the IP port information for each c= and m= line with information from the INVITE with Replaces: header. If there are more m= lines in the original offer than ports from the INVITE with Replaces: header, then the extra ports are disabled with port value of 0. If no SDP was offered in the original INVITE, then the SDP from the INVITE with Replaces: header is used as the offer in the 200 OK.

If the SDP media parameters were compatible between the replaced and replacing SDPs, then media does not need to be renegotiated and no re-INVITE is created. If the re-INVITE fails, the original dialogs are torn down using a BYE for the original server dialog.

INVITE with Replaces in Early Dialog Server Side

The Oracle Enterprise Session Border Controller does not support replacing an early server dialog. It replies with a 481 (Dialog/Transaction does not exist) response to the endpoint requesting the replace.



Replace Header Configuration

Replaces: header support is configured in the session-agent, realm, or sip-interface via the sip-profile configuration element. sip-profiles are defined once and attached to a chosen interface, realm or session-agent.

The replace-dialogs parameter is set to either enabled or disabled. In addition, you may set this parameter to inherit which uses the next lower order of precedence object. If there are no sip-profiles referenced in the higher ordered object, or if all the replace-dialogs parameters are set to inherit, then the feature is disabled.

To configure Replaces: header support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-profile and press Enter. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

```
ACMEPACKET (session-router)# sip-profile
ACMEPACKET(sip-profile)#
```

4. replace-dialogs—Set this parameter to enabled to enable Replaces: header support. A replaces parameter is also inservted in the Supported: header as sent into the realm where this sip profile is applied. You may also set this parameter to inherit for the element which this sip-profile is applied to, to inheret the value from the next-lower element.
5. Type done to save your work and continue.

Debugging

show sipd status

Includes Replaced Dialogs counts to show successfully replaced dialogs:

```
# show sipd status
17:41:38-142
SIP Status                -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Replaced Dialogs     -       -       1          1       1
```

show sipd errors

Includes Replace Dialog Fails to show failed dialog replacements. This counter is incremented only when the dialog replacement attempt actually occurred but failed to successfully complete.

```
# show sipd errors
17:58:04-181
SIP Errors/Events             ---- Lifetime ----
                        Recent      Total  PerMax
Replace Dialog Fails         0          0       0
```

SIP Options

This section explains how you can configure a limited list of specialized SIP features and/or parameters called options. The options described here were developed to meet specific needs not addressed by the standard SIP configuration parameters. Not all users have a need for these options.

Note: Oracle recommends checking with your Oracle representative before applying any of these options.

Overview

You can configure options for the SIP configuration and SIP interface. Both elements include a parameter (options) that you use to configure the options.

Global SIP Options

The following table lists the SIP options supported by the Oracle Enterprise Session Border Controller (E-SBC).

Option	Description
add-error-to-tag=no	If present (even when set to no), suppresses the addition of an Acme tag on 3xx-6xx responses.
add-prov-to-tag=no	Prevents the E-SBC from adding a tag parameter to the To header (to-tag) to non-100 provisional responses to INVITE requests. Used when a provisional (101-199) response is received from the UAS on a client transaction without a to-tag. By default, the E-SBC adds the tag cookie

Option	Description
	in the response (as though it had a tag) sent back to the UAC for the associated server transaction. When you include this option in the SIP configuration, and the response from the UAS does not have a to-tag, the response forwarded to the UAC will not have a to-tag.
add-reg-expires	Causes an Expires header to always be included in a REGISTER response with the registration caching and HNT traversal functions of the E-SBC. Use for endpoints that do not understand the Expires parameter in the Contact header.
add-ruri-user=<methods>	Causes a userinfo portion to be added to a Request-URI when one is not present. Used to support the OKI phone, which registers a Contact of just an IP-Address but rejects initial INVITEs if the Request_URI does not have a userinfo part. <methods> is a comma-separated list of methods to which the option should apply. If more than one method is listed, the list must be enclosed in quotes. This option only applies to out-of-dialog requests (no tag parameter in the To header). However, if ACK is listed, it will apply to all ACK requests because an ACK is always supposed to have a to-tag.
allow-notify-no-contact	Prevents the E-SBC from rejecting NOTIFYs with a 400 Bad Request response.NOTIFY requests without Contact header are allowed to pass through the E-SBC instead.
call-id-host=<host>	Causes the E-SBC to include a host part (ID@host) in the Call-ID it generated. <host> is the hostname (or IP address) that is to appear in the host part of the Call-ID. If not specified, the SIP port address is used.
contact-endpoint=<param-name>	Defines a URL parameter to report the real Contact address of an endpoint in a REGISTER message forwarded to a registrar, when the E-SBC is caching registration. (plain or HNT). If <param-name> is not specified, the default value endpoint is used. This parameter is added as a URL parameter in the Contact on the REGISTER message. In order for the registration cache to work properly, the softswitch/ registrar is expected to include the endpoint parameter in the Request-URI of a SIP request it forwards to the address-of-record.
contact-firewall=<param-name>	Defines a URL parameter to report the NAT between the E-SBC and the real Contact address of an endpoint in a REGISTRAR message forwarded to a registrar when the E-SBC is doing registration caching for NHT. If <param-name> is not specified, the default value firewall is used. This parameter will be added as a URL parameter in the Contact on the REGISTER message. In order for the registration cache to work properly, the softswitch/ registrar is expected to include the endpoint parameter in the Request-URI of any SIP request it forwards for the address-of-record.
disable-privacy	Prevents the change of the P-Preferred-Identity to P-Asserted-Identity and lets the P-Preferred-Identity go through unchanged.

Option	Description
drain-sendonly	Causes the E-SBC to examine the SDP attributes and change sendonly mode to sendrecv. This causes the endpoint receiving the SDP to send RTP, which is required for HNT traversal endpoints to work with media servers. The E-SBC sets up the flow so that RTP coming from the endpoint are dropped to prevent the UA that sent the sendonly SDP from receiving packets. See the option video-sbc-session also.
encode-contact=<prefix>	Causes the E-SBC to encode Contact addresses into the userinfo part of the URI. It applies only to Contact address that usually get the maddr parameter. Use when the E-SBC needs requests sent to the URI in the Contact sent instead to the E-SBC. The host part of the URI will have the E-SBC's address. The <prefix> serves as a place between the original userinfo and the encoded address. If a <prefix> is specified, a default of +SD is used. Without this option, the E-SBC adds a maddr parameter.
fix-to-header	For requests that have the E-SBC address in both the Request-URI and the To-URI, it sets the hostport of the To-URI to a local policy's next hop target on out-of-dialog requests (no to-tag). This is the default IWF behavior, even without this option configured.
forward-reg-callid-change	Addresses the case when an endpoint reboots and performs a third party registration before its old registration expires. During this re-registration, the contact header is the same as it was pre-reregistration. As a consequence of the reboot, the SIP Call-ID changes. In this situation, the E-SBC does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI. To remedy this problem, the E-SBC now keeps track of the Call-ID in its registration cache. A new option in the SIP interface configuration element forces the E-SBC to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.
global-contact	Addresses interoperability in the Dialog and Presence event packages that are used in hosted PBX and IP Centrex offerings. This option enables persistent URIs in the Contact headers inserted into outgoing SIP messages. If this option is not used, URIs placed in the Contact header of outgoing messages are only valid within the context of the dialog to which the message is associated.
ignore-register-service-route-oos	Prohibits a Register message from using a service route if that service route is an out-of-service session agent.
load-limit=<cpu percentage>	Defines the CPU usage percentage at which the E-SBC should start rejecting calls. Default value is 90%.
lp-sa-match=<match strategy>	Changes the ways local policies and session agents match; accounts for realm in matching process. Strategy choices are: all, realm, sub-realm, interface, and network.

Option	Description
max-register-forward=<value>	Defines a limit (as assigned in the value field) of REGISTERs to be forwarded to the registrar. During each second, the sipd counts how many REGISTERs have been sent to the registrar. It checks the threshold when it receives a REGISTER from the UA and determines that less than half the real registration lifetime is left. If the number of REGISTERs forwarded (new and updates) in the current second exceeds the configured threshold, it will respond to the UA from the cache.
max-register-refresh=<value>	Defines the desired limit of REGISTER refreshes from all the UAs. Each second of time, sipd counts the number of REGISTER/200-OK responses sent back. When the threshold is exceeded, it increments the expire time (based on NAT interval) by one second and resets the count. By default no threshold is applied. The recommended value is somewhat dependent on the E-SBC hardware used, but 300 can be used as an initial value.
max-routes=<number of routes>	Restricts the number of routes through which the sipd will iterate from a local policy lookup. For example, setting this option to 1 causes the E-SBC to only try the first, best, route. Setting this option to 0, or omitting it, lets the E-SBC use all of the routes available to it (with the priority scheme for route matching). When you test a policy using the test-policy ACLI command, this option is not recognized and all options that match the criteria are displayed.
max-udp-length=<maximum length>	Setting this option to zero (0) forces sipd to send fragmented UDP packets. Using this option, you override the default value of the maximum UDP datagram size (1500 bytes; sipd requires the use of SIP/TCP at 1300 bytes). You can set the global SIP configuration's max-udp-length=x option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.
media-release=<header-name>[;<header-param>]	Enables the multi-system media release feature that encodes IP address and port information for the media streams described by SDP. It lets another E-SBC decode the data to restore the original SDP, which allows the media to flow directly between endpoints in the same network (that is serviced by multiple E-SBCs). The media release information can appear in the following places: • SIP header P-Media-Release: <encoded-media-interface-information> • Header parameter on a SIP header Contact: <sip:1234@abc.com> ; acme-media=<encoded-media-interface-information> • SDP attribute in the message body a=acme-media: <encoded-media-interface-information> Option includes the following:

Option	Description
	• <header-name> is SIP header in which to put the information or the special value sdp, which indicates the information should be put into the SDP. • <header-param> is the header parameter name in which to put the information or in the case of the special header name value sdp, it is the SDP attribute name in which to put the information. They identify to where the encoded information is passed. If you do not specify a header, P-Media-Release is used.
no-contact-endpoint-port	Enables the E-SBC to add a URL parameter (defined as an argument to the contact-endpoint option) to the Contact headers of REGISTER messages that it forwards to the registrar when it performs registration caching. The value of the contact-endpoint URL parameter is the real address of the endpoint; and if the endpoint is behind a NAT, this includes the IP address and a port number. However, not all network entities can parse that port number, which is included unconditionally. This feature allows you to configure the exclusion of the port number. Despite the fact that you set this parameter in the global SIP configuration, it is applied only to SIP interfaces. However, you can set a contact-endpoint option in the realm configuration, on which this new parameter has no effect.
refer-to-uri-prefix=<prefix>	Defines a prefix to be matched against the userinfo part of Contact headers (config=), of which the E-SBC should create a B2BUA map. This ensures that outgoing messages include the correct userinfo value. This option is used to enable add-on conferencing.
reg-cache-mode=<mode>	Affects how the userinfo part of Contact address is constructed with registration caching. <mode> values are: • none: userinfo from the received (post NAT) Contact is retained • from: userinfo from the From header is copied to the userinfo of the forwarded Contact header • append: append the UA's Contact address into a cookie appended to the userinfo from the original Contact userinfo. For HNT, the NAT/firewall address is used. • append-from: takes userinfo from the From header and appends the encrypted address to the userinfo from the original Contact userinfo. For HNT, the NAT/firewall address is used. The from mode is used with softswitches that do not use the cookies used by the E-SBC. It also helps limit the number of bytes in the userinfo; which might create duplicate contacts. For example, if the E-SBC address is 1.2.3.4, both 1234@5.6.7.8 and 1234@4.3.2.1 will result in a E-SBC contact of 1234@5.6.7.8.
reg-contact-user-random	Support the SIP random registered-contact feature. Gives the E-SBC the ability to support endpoints that randomly change their contact usernames every time they re-register. Only applicable to operators who need to support the Japan TTC standard JJ-90.22 in specific applications. Applies to cases when an endpoint re-registers with a different contact username, but with the same hostname/IP address and the same address

Option	Description
	of record (AoR). Without this feature enabled, the E-SBC forwards every re-registration to the registrar with the new contact information without it being considered a registration refresh. The E-SBC forwards it to the Registrar using the same sd-contact as the previous registration. When you set this option, the E-SBC does treat such a re-registration as a registration refresh when it is received prior to the half-life time for the specific contact. The E-SBC also uses the new contact username for the Request-URI in requests it sends to the UA, and verifies that the UA uses the correct one when that E-SBC is set to allow-anonymous registered mode. NOTE: The registration cache mode is set using the option reg-cache-mode, but regardless of how you configure it, the registration cache mode will be set to contact when SIP random registered-contact feature is enabled.
register-grace-timer	Makes the grace time for the SIP Registration configurable. You can configure the grace timer in seconds.
reinvite-trying=[yes]	Causes the E-SBC to send a 100 Trying for re-INVITEs, which is normally suppressed. If you enter the option name but omit the value yes, the option is still active.
reject-interval=<value>	Acts as a multiplier to increase the value presented to the UAC in the Retry-After field. For example, if reject-interval=5 (reject interval is set to 10); at a 90% rejection rate the E-SBC sends Retry-After: 45. When rejecting calls because of CPU load limiting, the E-SBC adds a Retry-After parameter to the error response (typically 503 Service Unavailable). By default the E-SBC sets the Retry-After value to be 1/10th of the current rejection rate.
reject-register=[no | refresh]	Allows REGISTER messages through even during load limiting. By default, REGISTER messages are subject to load limiting.
response-for-not-found=<response code>	Change the 404 Not Found generated by the E-SBC to a different response code.
route-register-no-service-route	Controls how a UA is registered. Option can have three values: • route-register-no-service-route—This option prevents the use of the Service-Route procedure to route the Re-Register requests after the UA has initially registered. • route-register-no-service-route=all—Prevents the use of the Service-Route procedure to route the Re-Register requests for all messages, after the UA has initially registered. • route-register-no-service-route=refresh—Prevents the use of the Service-Route procedure to route the Re-Register requests for all refresh-register messages, but not de-register messages, after the UA has initially registered. Addition idle argument ensures that, when enabled, the E-SBC follows the previously defined rules for idle calls, where idle means not engaged in any INVITE-based sessions. Sample syntax: route-register-no-service-route=refresh;idle

Option	Description
sdp-insert-sendrecv	When a call is initiated, the SDP communicates between call offerer and call answerer to determine a route for the media. Devices can be configured to only send media (“a=sendonly”), to only receive media (“a=recvonly”), or to do both (“a=sendrecv”). Some devices, do not disclose this information. With this option configured, when either the offerer or answerer does not disclose its directional attribute, the E-SBC automatically inserts a sendrecv direction attribute to the media session.
set-inv-exp-at-100-resp	Set Timer C when a 100 Trying response is received (instead of waiting until 1xx (> 100) is received). If the E-SBC does not receive a 100 Trying response within Timer B, the call should be dropped because there is a problem communicating with the next hop.
strip-domain-suffix-route	Causes sipd to strip any Router headers from the inbound messages coming to the external address of a SIP NAT; if the message contains a FQDN that matches the configured domain suffix for that SIP NAT.
video-sbc-session	Use with drain-sendonly for conference floor support. When configured with drain-sendonly and when the E-SBC receives an SDP, the E-SBC proxies the m=control and its related a= and c= unchanged. Although media streams are allocated for this m line, an actual flow is not set up. SDP received with the following: m=video a=sendonly is sent out as the following: m=video a=sendonly a=X-SBC-Session
session-timer-support	This option enables the E-SBC to start the session timer for session refreshes coming from the UAC. The E-SBC determines whether or not a session is active based on session refreshes or responses. It terminates the session when no session refreshes occur within the session timer interval.
session-timer-support	Enables RFC4028 session timer support.
inmanip-before-validate	Enables SIP Header Pre-processing for HMR.
proccess-implicit-tel-URI	Correctly appends coodie in REGISTER message when user=phone does not exist.
offerless-bw-media	Reserves appropriate bandwidth for an INVITE with no SDP.

SIP Interface Options

The following table lists the SIP interface options supported by the Oracle Enterprise Session Border Controller.

Option	Description
100rel-interworking	Enables RFC 3262, *Reliability of Provisional Responses in the Session Initiation Protocol* support.

Option	Description
contact-endpoint=<endpoint name>	The Oracle Enterprise Session Border Controller inserts the endpoint IP address and port into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value endpoint is used.
contact-firewall=<firewall name>	The Oracle Enterprise Session Border Controller inserts the firewall IP address and port into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value firewall is used.
contact-vlan=<VLAN/realm name>	The Oracle Enterprise Session Border Controllerinserts the realm and VLAN ID into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value vlan is used.
dropResponse	The Oracle Enterprise Session Border Controller drops responses by specified status codes. The option value can contain one or more status codes separated by semicolons. Error ranges can also be entered. If any of the response codes matches then a response is not sent. If the dropResponse option is set in both the sip-interface and the session-agent elements, the session-agent setting takes precedence.
max-udp-length=<maximum length>	Sets the largest UDP packers that the Oracle Enterprise Session Border Controller will pass. Packets exceeding this length trigger the establishment of an outgoing TCP session to deliver the packet; this margin is defined in RFC 3261. The system default for the maximum UDP packet length is 1500. You can set the global SIP configuration's max-udp-length=x option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.
response-for-not-found=<response code>	Change the 404 Not Found generated by the SBC to a different response code.
strip-route-headers	Causes the Oracle Enterprise Session Border Controller to disregard and strip all route headers for requests received on a SIP interface.
upd-fallback	When a request needs to be sent out on the SIP interface for which you have configured this option, the Oracle Enterprise Session Border Controller first tries to send it over TCP. If the SIP endpoint does not support TCP, however, then the Oracle Enterprise Session Border Controller falls back to UDP and tries the request again.
via-header-transparency	Enables the Oracle Enterprise Session Border Controller to insert its Via header on top of the top-most Via header received from user equipment (UE). It then forwards it on to the IP Multimedia Subsystem (IMS) core with the original Via header now located as the bottom-most Via header.

Option	Description
	The Oracle Enterprise Session Border Controller still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.
use-redirect-route	Use Route parameter in Contact header as next-hop as received in a 3xx response.
reg-via-proxy	Enables your Oracle Enterprise Session Border Controller to support endpoints that register using an intervening proxy.
lmsd-interworking	Enables 3GPP2 LMSD Interworking.
suppress-reinvite	Enables reINVITE supression.

SIP Session Agent Options

The following table lists the SIP session agent options supported by the Oracle Enterprise Session Border Controller.

Option	Description
dropResponse	The Oracle Enterprise Session Border Controllerdrops responses by specified status codes. The option value can contain one or more status codes separated by semicolons. Error ranges can also be entered. If any of the response codes matches then a response is not sent. If the dropResponse option is set in both the sip-interface and the session-agent elements, the session-agent setting takes precedence.
trans-timeouts=<value>	Defines the number of consecutive non-ping transaction timeouts that will cause a session agent to be out of service. When the session agent is configured, i.e. when the PING options are defined, the value is 10. If not defined, the default value is 5. A Value of 0 prevents the session agent from going out of service because of a non-ping transaction timeout.
via-origin=<parameter-name>	Causes a parameter to be included in the top Via header of requests sent to the session agent. The parameter indicates the source IP address of the corresponding request received by the Oracle Enterprise Session Border Controller. <parameter-name> defines the name of the parameter. If not specified, the default value origin is used.
refer-reinvite	Enables SIP REFER with Replaces.

SIP Realm Options

The following table lists the SIP session agent options supported by the Oracle Enterprise Session Border Controller.

Option	Description
number-normalization	Applies to the SIP To URI. (Currently the Oracle Enterprise Session Border Controller supports number normalization on From and To addresses for both inbound and outbound call legs.) Number normalization includes add, delete, and replace string functions that result in consistent number formats. Number normalization occurs on ingress traffic, prior to the generation of accounting records or local policy lookups. (also applies for H.323 to SIP calls.)

Option	Description
refer-reinvite	Enables SIP REFER with Replaces.

SIP Realm Options Configuration

To configure options:

Labels enclosed in <> indicate that a value for the option is to be substituted for the label. For example, <value>. In order to change a portion of an options field entry, you must re-type the entire field entry.

1. Navigate to the options parameter in the SIP configuration or SIP interface elements.
2. Enter the following:

   ```
   options Space <option name>=”<value>”
   ```

   For example, if you want to configure the refer-to-uri-prefix option (the add-on conferencing feature):

   Type options, followed by a Space.

   Type refer-to-uri-prefix, followed by an equal sign (=).

   Type the opening quotation mark (“) followed by conf, another equal sign and the closing quotation mark.

   Press Enter.

   For example:

   ```
   options refer-to-uri-prefix=conf=
   ```

   If the feature value itself is a comma-separated list, it must be enclosed in quotation marks.

Configuring Multiple Options

You can enter a list of options for this field:

1. Type options followed by a space.
2. Within quotation marks, enter the feature names and values of the parameters you need. Separate each one with a comma.
3. Close the quotation marks.
4. Press Enter.

   For example:

   ```
   ACMEPACKET(sip-config)# options “refer-to-uri-prefix=“conf=”,encode-
   contact=“+SD”,add-ruri-user=INVITE,ACK”
   ```

Adding an Entry

Enter the new entry with a preceding plus (+) sign. For example:

```
options +response-for-not-found
```

This format allows previously configured options field values to remain intact without requiring re-entry of the entire field value.

SIP Security

This section provides an overview of Oracle Enterprise Session Border Controller’s security capability. Oracle Enterprise Session Border Controller security is designed to provide security for VoIP and other multi-media services. It includes access control, DoS attack, and overload protection, which help secure service and protect the network infrastructure (including the Oracle Enterprise Session Border Controller). In addition, Oracle Enterprise Session Border Controller security lets legitimate users to still place calls during attack conditions, protecting the service itself.

Oracle Enterprise Session Border Controller security includes the Net-SAFE framework's numerous features and architecture designs. Net-SAFE is a requirements framework for the components required to provide protection for the Session Border Controller (SBC), the service provider's infrastructure equipment (proxies, gateways, call agents, application servers, and so on), and the service itself.

Denial of Service Protection

The Oracle Enterprise Session Border Controller Denial of Service (DoS) protection functionality protects softswitches and gateways with overload protection, dynamic and static access control, and trusted device classification and separation at Layers 3-5. The Oracle Enterprise Session Border Controller itself is protected from signaling and media overload, but more importantly the feature allows legitimate, trusted devices to continue receiving service even during an attack. DoS protection prevents the Oracle Enterprise Session Border Controller host processor from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Volume-based attack (flood) of valid or invalid call requests, signaling messages, and so on.
- Overload of valid or invalid call requests from legitimate, trusted sources

Levels of DoS Protection

The multi-level Oracle Enterprise Session Border Controller Denial of Service protection consists of the following strategies:

- Fast path filtering/access control: involves access control for signaling packets destined for the Oracle Enterprise Session Border Controller host processor as well as media (RTP) packets. The SBC accomplishes media filtering using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted list and a deny list for each Oracle Enterprise Session Border Controller realm (although the actual devices can be dynamically trusted or denied by the Oracle Enterprise Session Border Controller based on configuration). You do not have to provision every endpoint/device on the Oracle Enterprise Session Border Controller, but instead retain the default values.
- Host path protection: includes flow classification, host path policing and unique signaling flow policing. Fast path filtering alone cannot protect the Oracle Enterprise Session Border Controller host processor from being overwhelmed by a malicious attack from a trusted source. The host path and individual signaling flows must be policed to ensure that a volume-based attack will not overwhelm the Oracle Enterprise Session Border Controller's normal call processing; and subsequently not overwhelm systems beyond it. The Oracle Enterprise Session Border Controller must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent. At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Oracle Enterprise Session Border Controller maintains two host paths, one for each class of traffic (trusted and untrusted), with different policing characteristics to ensure that fully trusted traffic always gets precedence.
- Host-based malicious source detection and isolation – dynamic deny list. Malicious sources can be automatically detected in real-time and denied in the fast path to block them from reaching the host processor.

Configuration Overview

NAT table entries are used to filter out undesired IP addresses (deny list). After the packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager based on the sender's IP address. NAT table entries are used to distinguish signaling packets coming in from different sources for policing purposes.

You can configure deny rules based on the following:

- ingress realm
- source IP address
- transport protocol (TCP/UDP)
- application protocol (SIP, MGCP)

You can configure guaranteed minimum bandwidth for trusted and untrusted signaling paths.

You can configure signaling path policing parameters for individual source addresses. Policing parameters include:

- peak data rate in bits per second
- average data rate in bits per second
- maximum burst size

SIP Unauthorized Endpoint Call Routing

The Oracle Enterprise Session Border Controller (E-SBC) can route new dialog-creating SIP INVITEs from unauthorized endpoints to a session agent or session agent group; then rejection can occur based on the allow-anonymous setting for the SIP port. This type of provisional acceptance and subsequent rejection applies only to INVITEs; the E-SBC continues to reject all other requests, such as SUBSCRIBE.

You might enable this feature if you have a network in which unauthorized SIP endpoints continually try to register even if the Oracle Enterprise Session Border Controller has previously rejected them and never will accept them. For instance, the user account associated with the endpoint might have been removed or core registrars might be overloaded.

SIP Unauthorized Endpoint Call Routing Configuration

You enable the routing of unauthorized endpoints to session agents and session agent groups that will reject them in the SIP interface configuration.

To enable SIP unauthorized endpoint call routing:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-interface and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. route-unauthorized-calls—Enter the name (or IP address) of the session agent or session agent group to which you want calls from unauthorized endpoints routed. This parameter is blank by default, meaning the SIP unauthorized call routing feature is disabled.

   Remember your settings in the allow-anonymous parameter in the SIP port configuration provide the basis for rejection.

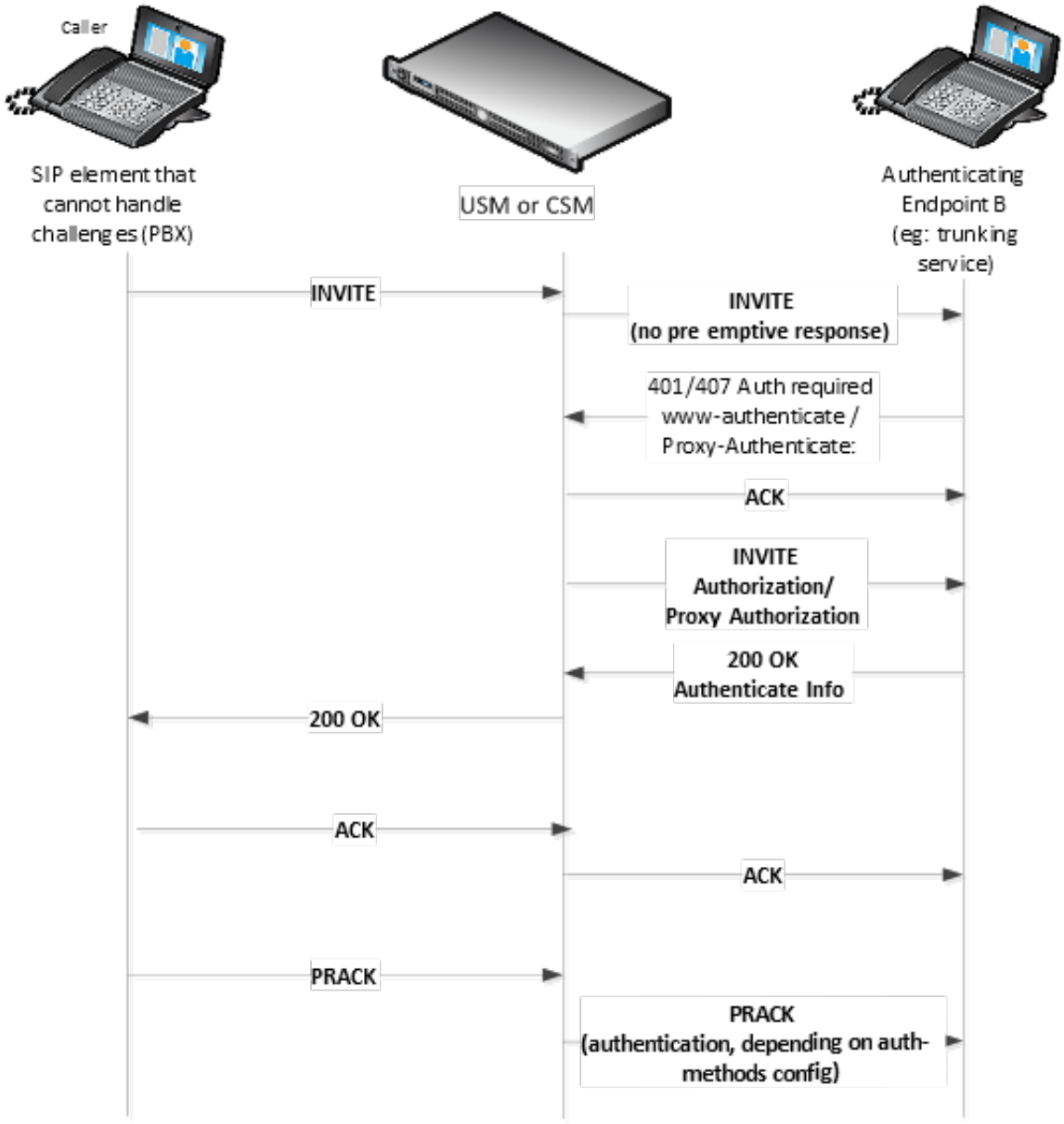
5. Save and activate your configuration.

Digest Authentication with SIP

Digest authentication for Session Initiation Protocol (SIP) is a type of security feature on the Oracle Enterprise Session Border Controller that provides a minimum level of security for basic Transport Control Protocol (TCP) and User Datagram Protocol (UDP) connections. Digest authentication verifies that both parties on a connection (host and endpoint client) know a shared secret (a password). This verification can be done without sending the password in the clear.

Digest authentication is disabled by default on the Oracle Enterprise Session Border Controller. When digest authentication is enabled, the Oracle Enterprise Session Border Controller (host) responds to authentication challenges from SIP trunking Service Providers (endpoint client). The Oracle Enterprise Session Border Controller performs authentication for each IP-PBX initiating the call. However, the authentication challenge process takes place

between the host and the client only since the IP-PBX cannot handle authentication challenges. The following illustration shows the digest authentication process.



The digest authentication scheme is based on a simple challenge-response paradigm. A valid response contains a checksum (by default, the MD5 checksum) of the “username” and password. In this way, the password is never sent in the clear.

By default, the Oracle Enterprise Session Border Controller uses cached credentials for all requests within the same dialog, once the authentication session is established with a 200OK from the authenticating SIP element. If the in-dialog-methods attribute contains a value, it specifies the requests that have challenge-responses inserted within a dialog.

In digest authentication with SIP, the following can happen:

- More than one authenticating SIP element (IP-PBX) may be the destination of requests.

- More than one authentication challenge can occur in a SIP message. This can occur when there are additional authenticating SIP elements behind the first authenticating SIP element.
- The Oracle Enterprise Session Border Controller distinguishes whether the IP-PBX is capable of handling the challenge. If Digest Authentication is disabled (no auth-attributes configured) on the Session Agent, the challenge is passed back to the IP-PBX.

Note: If there are multiple challenges in the request, and if the Oracle Enterprise Session Border Controller has only some of the cached credentials configured, the Oracle Enterprise Session Border Controller adds challenge-responses for the requests it can handle, and does not pass the challenge back to the IP-PBX.

Challenge-Responses in Requests not in the Dialog

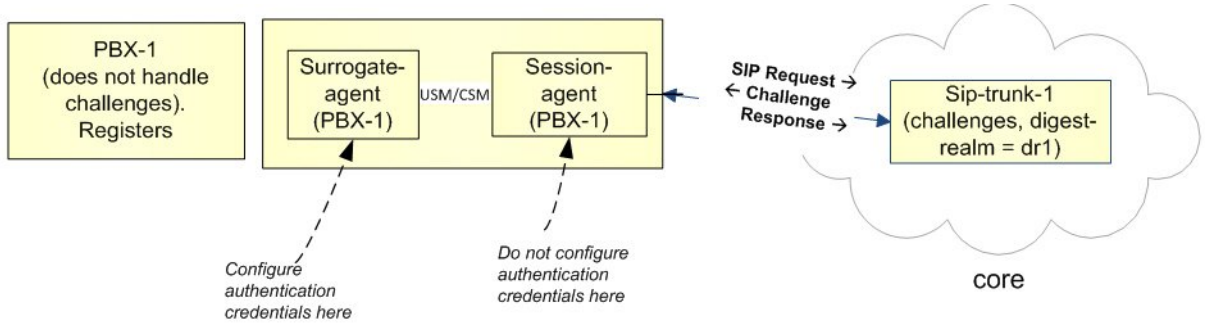
A digest authentication session starts from the client response to a www-authenticate/proxy-authenticate challenge and lasts until the client receives another challenge in the protection space defined by the auth-realm. Credentials are not cached across dialogs; however, if a User Agent (UA) is configured with the auth-realm of its outbound proxy, when one exists, the UA may cache credentials for that auth-realm across dialogs.

Note: Existing Oracle Enterprise Session Border Controller behavior with surrogate-agents is that they cache credentials from REGISTER for INVITE sessions only if the Oracle Enterprise Session Border Controller is considered a UA sending to its outbound proxy.

Surrogate Agents and the Oracle Enterprise Session Border Controller

In the case where a surrogate-agent is configured for the IP-PBX, you do not have to configure digest authentication attributes in the session-agent object for the same IP-PBX. The surrogate-agent authentication configuration takes precedence over the session-agent authentication configuration and so it is ignored.

The following illustration shows an example of a surrogate-agent with a session-agent in the network.



Configuring Digest Authentication

In the Oracle Enterprise Session Border Controller ACLI, you can access the Digest Authentication object at the path session-router->session-agent->auth-attribute. If enabled, the Digest Authentication process uses the attributes and values listed in this table.

Note: If enabling Digest Authentication, all attributes listed below are required except for the in-dialog-methods attribute which is optional.

The following table lists the digest authentication object

```
ACMEPACKET(auth-attribute)# show
        auth-attribute
                auth-realm                      realm01
                username                        user
                password                        ********
                in-dialog-methods               ACK INVITE SUBSCRIBE
```

To configure digest authentication on the Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter to access the session agent-related attributes.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Type auth-attribute and press Enter to access the digest authentication-related attributes.

```
ACMEPACKET(session-agent)# auth-attribute
ACMEPACKET(auth-attribute)#
```

5. auth-realm — Enter the name (realm ID) of the host realm initiating the authentication challenge. This value defines the protected space in which the digest authentication is performed. Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# auth-realm realm01
```

6. username — Enter the username of the client. Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# username user
```

7. password — Enter the password associated with the username of the client. This is required for all LOGIN attempts. Password displays while typing but is saved in clear-text (i.e., *****). Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# password *******
```

8. in-dialog-methods — Enter the in-dialog request method(s) that digest authentication uses from the cached credentials. Specify request methods in a list form separated by a space enclosed in parentheses. Valid values are:

- INVITE | BYE | ACK | CANCEL | OPTIONS | SUBSCRIBE | PRACK | NOTIFY | UPDATE | REFER

```
ACMEPACKET(auth-attribute)# in-dialog-methods (ack invite subscribe)
```

Note: The methods not in this list are still resubmitted if a 401/407 response is received by the Oracle Enterprise Session Border Controller.

If you do not specify any in-dialog-method value(s), digest authentication does not add challenge-responses to in-dialog requests within a dialog.

This attribute setting applies to in-dialog requests only.

Additional Notes

The following are additional notes that describe the digest authentication process:

- The Oracle Enterprise Session Border Controller always challenges the first LOGIN request, and initial authentication begins with that request. The recalculated authorization key — the credentials — are then included in every subsequent request.
- If the Oracle Enterprise Session Border Controller does not receive any communication from the client within the expiration period, the Oracle Enterprise Session Border Controller logs the client out and tears down the transport connection. Faced with interface loss, the Oracle Enterprise Session Border Controller default behavior is to flush all warrant information from the target database. This response necessitates that the client first login/re-register with the Oracle Enterprise Session Border Controller, and then repopulate the empty database using a series of ADD requests. This behavior ensures that client and Oracle Enterprise Session Border Controller target databases are synchronized.

  Alternatively, when faced with interface loss, the Oracle Enterprise Session Border Controller can retain all warrant information within the target database. This response necessitates only that the client first login/re-register with the Oracle Enterprise Session Border Controller. After successful registration the client should, but is not

required to, use a series of GET, ADD, and DELETE requests to ensure that the Oracle Enterprise Session Border Controller and client target databases are synchronized.

- The Oracle Enterprise Session Border Controller ignores the Authentication-Info header that comes in the 200OK response after digest authentication is complete. The Oracle Enterprise Session Border Controller receives a 401/407 response from the client. However, some surrogate-agents may process the Authentication-Info header in a single challenge.

Digest Authentication and High Availability

The Oracle Enterprise Session Border Controller supports digest authentication in high availability (HA) environments. The session-agent configuration, which includes the digest authentication parameters on the primary Oracle Enterprise Session Border Controller, are replicated on the HA Oracle Enterprise Session Border Controller. However, cached credentials on the primary device are not replicated on the HA device.

SIP NAT Function

This section explains how to configure the optional SIP NAT function. You can configure the SIP NAT function if you need to translate IP address and UDP/TCP port information. The SIP NAT function also prevents private IP addresses in SIP message URIs from traveling through an untrusted network.

Overview

TheOracle Enterprise Session Border Controller is an intermediary device that provides NAT functions between two or more realms. It translates IP addresses between untrusted and trusted networks using NAT. A trusted network is inside the NAT, and a untrusted network is outside the NAT. A NAT also lets a single IP address represent a group of computers.

For SIP, the SIP NAT function on the Oracle Enterprise Session Border Controller does the following:

- routes SIP packets between the Oracle Enterprise Session Border Controller's SIP proxy (B2BUA) and external networks (or realms), including the translation of IP address and UDP/TCP port information.
- prevents private IP addresses in SIP message URIs from traveling through the untrusted network. SIP NAT either translates the private address to one appropriate for an untrusted address or encrypts the private address into the URI.

Packets arriving on the external address (at port 5060) are forwarded to the Oracle Enterprise Session Border Controller's SIP proxy with the source address changed to the home address (at port 5060). When the Oracle Enterprise Session Border Controller's SIP proxy sends packets to the home address (at port 5060), they are forwarded to the external proxy address (and external proxy port), with the source address changed to the external address (at port 5060).

Note: The SIP config's NAT mode parameter works in conjunction with the SIP NAT function configuration. It identifies the type of realm in which the SIP proxy is located (public or private) and affects whether IPvr addresses in SIP messages are encoded.

The translation of URIs in the actual SIP message occurs as messages are received and sent from the Oracle Enterprise Session Border Controller's SIP proxy. For the messages being sent to the external network, the contents of the SIP message are examined after the translation to determine if the destination needs to be changed from the external proxy address to an address and port indicated by the SIP message. This process takes place so the request is sent to where the Request-URI or the Route header indicates, or so the response is sent to where the Via indicates.

NAT Modes

The specific addresses used in translating URIs in the SIP message depend on whether the Oracle Enterprise Session Border Controller is performing NAT functions for a trusted or untrusted network. This condition is determined by the NAT mode value you enter when you configure the SIP config element. The NAT modes are:

- untrusted—The SIP proxy is associated with an address for an untrusted network (the address value you entered when you configured the SIP interface's SIP port parameter), and the home address in the SIP NAT is the address

of the external realm/network. When the URI contains the external address, it is translated to the SIP NAT's home proxy address (or to the SIP port address if the home proxy address field is empty). When a URI contains the external proxy address, it is translated to the home address.

If the URI contains any other private address (matching the realm's address prefix, identified in the SIP NAT's realm ID), it is encrypted and the address is replaced with the home address value. If the URI contains a user part, a suffix consisting of the user NAT tag and the encrypted address is appended to the user part. For example, with a user NAT tag value of -private-, the private URI of sip@123192.169.200.17:5060 will become the public URI of sip:123-private-eolmhet2chbl3@172.16.0.15.

If there is no user part, the host consists of the host NAT tag followed by the encrypted address and the domain suffix. A maddr parameter equal to the home address (or received in the case of a Via header) is added to the URI. For example, with a host NAT tag value of PRIVATE- and a domain suffix value of private.com, the private URI of sip:192.168.200.17:5060 will become the public URI of sip:PRIVATE-eolmhet2chbl3.private.com:5060;maddr=172.16.0.15.

- trusted—The SIP proxy is on a trusted network (the address value you entered when you configured the SIP interface's SIP port parameter), and the SIP NAT's external address is the public address of the external realm/network. When the URI contains the home address value, it is translated to the value set for the external proxy address. When the URI contains the SIP proxy's address, it is translated to the external address. If the URI contains any other private address (matching the realm's address prefix, identified in the SIP NAT's realm ID), the private address is encrypted and the address is replaced with the external address.

  Note: Do not use the home proxy address value with private NAT functioning.

Adding a maddr Parameter to a URI

When you configure a SIP interface, you can configure the contact mode. The contact mode sets the contact header routing mode, which determines how the contact address from a trusted network is formatted. You set the contact mode to add a maddr parameter equal to the SIP proxy to the URI in the Contact header. For example, the URI from the prior example (sip:192.168.200.17:5060) becomes sip:123-trusted-eolmhet2chbl3@172.16.0.15;maddr=172.16.0.12.

Note: For SIP elements that do not support the maddr parameter, configure a Contact mode as none.

You might require this encryption to cause other SIP elements in the untrusted network to send requests directly to the SIP proxy. Otherwise, the requests are sent to the home address. However, responses sent by the SIP proxy will have the SIP proxy's source address, rather than the home address. Some SIP elements might drop responses that come from a IP address different from the one to which the request is sent.

About Headers

You can specify which SIP headers you want effected by the SIP NAT function. The URIs in these headers are translated and encrypted, the encryption occurs according to the rules of this SIP NAT function.

You can enter header values by using either the full header name or its corresponding abbreviation, if applicable. The following table lists the available headers and their corresponding abbreviations

Header	Abbreviation
Call-ID	i
Contact	m
From	f
Record-Route	none
Route	none
Ready-To	none

Header	Abbreviation
Replaces	none
Refer-To	r
To	t
Via	v

SIP sessions are terminated and re-originated as new sessions as they are routed through the Oracle Enterprise Session Border Controller. Among the actions performed, SIP headers are modified to prevent the transmission of IP address and route information.

Replacing Headers

In the SIP signaling message, any Via headers are stripped out and a new one is constructed with the Oracle Enterprise Session Border Controller's IP address in the sent-by portion. If a Contact header is present, it is replaced with one that has the Oracle Enterprise Session Border Controller's IP address. All other headers are subject to NATing based on the following rules:

- The Request-URI is replaced with the next hop's IP or FQDN address.
- All other headers are replaced based on the two SIP NAT function SIP NAT function rules

Mapping FQDNs

The Oracle Enterprise Session Border Controller maps FQDNs that appear in the certain headers of incoming SIP messages to the IP address that the 12:00 inserts in outgoing SIP contact headers. The mapped FQDNs are restored in the SIP headers in messages that are sent back to the originator.

This feature is useful to carriers that use IP addresses in the SIP From address to create trunk groups in a softswitch for routing purposes. When the carrier's peer uses FQDNs, the carrier is forced to create trunk groups for each possible FQDN that it might receive from a given peer. Similarly, this can apply to SIP Contact and P-Asserted-Identity headers.

SIP NAT Function Cookies

Cookies are inserted to hide that information is coming from a realm external to the home realm. They are used when information needs to be placed into a given element of a SIP message that must also be seen in subsequent SIP messages within a flow. When forwarding a SIP message, the Oracle Enterprise Session Border Controller (E-SBC) encodes various information in the outgoing message, which is passed from one side to another in SIP transactions.

SIP NAT function cookies let the E-SBC hide headers, IPv4 addresses, and SIP URIs. These cookies are included when certain conditions are present in E-SBC SIP transactions.

Oracle's SIP NAT function cookies can be used in the userinfo, host, URL parameter, and tel URL parameter portions of the SIP message.

userinfo

The Oracle Enterprise Session Border Controller places a cookie in the userinfo portion of a SIP URI when a SIP header contains a SIP URI, and includes that header type in the list of headers to be hidden (encrypted) in the associated SIP NAT function. The cookie for the userinfo portion is the following:

```
[user nat tag][encrypted 13-byte host IP][encrypted 13 byte maddr IP (if
present)]
```

where:

- [user nat tag] refers to the SIP NAT function's original user NAT tag field.
- [encrypted 13-byte host IP] refers to the host IP encryption.
- [encrypted 13 byte maddr IP (if present)] refers to the maddr IP encryption, if it exists.

With a user NAT tag of -acme, the following SIP-URI:

```
sip:6175551212@192.168.1.100
```

might be translated into:

```
sip:6175551212-acme-pfi1s7n2pstna@172.16.1.10
```

Note: Multiple additional cookies might be appended with each hop (for example, from the external proxy to the home proxy and back).

host

When hiding IP addresses in a SIP message, the SIP NAT function generates the following cookie for a SIP-URI with no userinfo portion:

```
[host nat tag][encrypted 13-byte host IP][encrypted 13 byte maddr IP (if
present)][domain suffix]
```

where:

- [host nat tag] refers to the SIP NAT function's host NAT tag.
- [encrypted 13-byte host IP] refers to the host IP encryption.
- [encrypted 13 byte maddr IP (if present)] refers to the maddr IP encryption, if it exists.
- [domain suffix] refers to the SIP NAT function's domain suffix field.

With a SIP NAT function's host tag of ACME- and a domain suffix of .acme.com, the following SIP header:

Via: SIP/2.0/UDP 192.168.1.100:5060

might be translated into the following:

```
Via: SIP/2.0/UDP ACME-pfi1s7n2pstna.acme.com
```

URL Parameter

If the SIP NAT function's use url parameter field has a value of from-to or all, the SIP NAT function places all cookies generated to hide SIP URIs in a custom tag appended to the header. Setting the use url parameter field to:

- from-to only affects the behavior of the SIP NAT function's cookies in the From and To headers.
- all affects all SIP headers processed by the SIP NAT function

The cookie is the following:

[;url-parameter]=[host nat tag][encrypted 13-byte host IP][encrypted 13-byte maddr IP]

where:

- [;url-parameter] refers to the SIP NAT function's parameter name field.

  This cookie type is associated with the all and from-to field value options of the SIP NAT function's use url parameter field.
- [host nat tag] refers to the SIP NAT function's host NAT tag field.
- [encrypted 13-byte host IP] refers to the host IP encryption.
- [encrypted 13 byte maddr IP (if present)] refers to the maddr IP encryption, if it exists.

With a host NAT tag of ACME- and a parameter name of acme_param, the following SIP-URI:

```
sip:6175551212@192.168.1.100
```

might be translated into the following:

```
sip:6175551212@172.16.1.10;acme_param=ACME-pfi1s7n2pstna.
```

tel URL

The SIP NAT function cookie is used when devices in your network are strict about the context portion of SIP messages regarding the conversion of tel URLs. This cookie for the tel URL parameter portion of a SIP message is the following:

```
tel URL parameter-[13-byte host IP][13 byte optional maddr IP]domain suffix
```

where:

- tel URL parameter refers to the SIP NAT function's use url parameter.

  This cookie type is associated with the use url parameter's phone field value for the SIP NAT.
- [13-byte host IP] refers to the host IP encryption.
- [13 byte optional maddr IP] refers to the maddr IP encryption, if it exists.
- domain suffix refers to the SIP NAT function's domain suffix field.

Configuration Overview

Configuring the SIP NAT function falls into two areas, the SIP NAT interface parameters and the SIP NAT policies.

SIP NAT Interface

The following tables lists the SIP NAT function interface parameters you need to configure on the Oracle Enterprise Session Border Controller (E-SBC).

Parameter	Description
realm ID	Name of the external realm. The realm ID must be unique; no two SIP NATs can have the same realm ID. This realm ID must also correspond to a valid realm identifier entered when you configured the realm.
external proxy address	IPv4 address of the SIP element (for example, a SIP proxy) in the external network with which the E-SBC communicates. Entries must follow the IP address format.
external proxy port	UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the E-SBC communicates. Minimum value is 1025, and maximum value is 65535. Default is 5060.
external address	IPv4 address on the media interface in the external realm. Enter a value that ensures any packet with an external address value as its destination address is routed to the E-SBC through the media interface connected to or routable from the external realm. Entries must follow the IP address format. To specify whether the external realm referenced in this field is private or public, configure the SIP config's NAT mode.
home address	IPv4 address on the media interface in the home realm. Enter a value that ensures any packet with a home address value as its destination address must be routed to the E-SBC through the media interface connected to or routable from the home realm. Entries must follow the IP address format. The value entered in this field must be different from the IP address value of the home realm's network interface element. The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you

Parameter	Description
	entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.
home proxy address	Sets the IP address for the home proxy (from the perspective of the external realm). By default, this field is empty. An empty home proxy address field value signifies that there is no home proxy, and the external address will translate to the address of the E-SBC 's SIP proxy. Entries must follow the IP address format.
home proxy port	Sets the port number for the home realm proxy. Value can be set to zero (0). Minimum is 1025 and maximum is 65535. Default is 5060.
route home proxy	Whether to route all inbound requests for the SIP NAT to the home proxy. enabled adds route if Request-URI is not the E-SBC disabled does not route inbound requests to the home proxy forced always adds route

SIP NAT Function Policies

The following tables lists the SIP NAT function policy parameters you need to configure on the Oracle Enterprise Session Border Controller (E-SBC).

Parameter	Description
domain suffix	Domain name suffix of the external realm. The domain name suffix refers to and must conform to the hostname part of a URI. In combination with the user NAT tag and host NAT tag values, this value is used to help the E-SBC identify an encoded URI that it needs to translate when moving between public and private realms. This suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is ACME-abc123 and the domain-suffix value is .netnetsystem.com, the resulting FQDN will be ACME-abc123.netnetsystem.com.
address prefix	Defines which IPv4 address prefixes from incoming messages require SIP-NAT encoding (regardless of the realm from which these messages came).
tunnel redirect	Controls whether Contact headers in a 3xx Response message received by the E-SBC are NATed when sent to the initiator of the SIP INVITE message.
use url parameter	Establishes whether SIP headers will use the URL parameter entered in the parameter name for encoded addresses that the SIP NAT function creates. Also, if SIP headers will be used, which type of headers will use the URL parameter. For example, all headers or just the From and To headers. Enumeration field.
parameter name	Indicates the name of the URL parameter when use url applies. This field value will be used in SIP NAT encoding addresses that have a use url parameter value of either from-to or all.

Parameter	Description
user NAT tag	Identifies the prefix used when an address is encoded into the username portion of user@host;name=xxxx; where name = parameter name. The user NAT tag values can consist of any characters that are valid for the userinfo part of a URI. In combination with the domain suffix and host NAT tag field values, this value is used to help the E-SBC identify an encoded URI that it needs to translate when moving between public and private realms.
host NAT tag	Identifies the prefix used when encoding an address into the hostname part of the URI or into a URL parameter. The host NAT tag values refer to domain labels and can consist of any characters that are valid for the hostname part of a URI. In combination with the domain suffix and user NAT tag values, this value is used to help the E-SBC identify an encoded URI that it needs to translate when moving between public and private realms.
headers	Lists the SIP headers to be affected by the E-SBC SIP NAT function. The URIs in these headers will be translated and encrypted, and encryption will occur according to the rules of this SIP NAT.

SIP NAT Function Configuration

To configure the SIP NAT function on an Oracle Enterprise Session Border Controller (E-SBC):

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-nat and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-nat
ACMEPACKET(sip-nat)#
```

4. realm-ID—Enter the name of the realm you want to identify as the external realm.

   The name you use as the realm ID must be unique. No two SIP NAT functions can have the same realm ID. Also, this value must correspond to a valid identifier entry already configured for the realm.

5. domain-suffix—Enter the domain suffix to identify the domain name suffix of the external realm. The domain suffix must begin with a (.) dot.

   The domain name suffix refers to and must conform to the hostname part of a URI. For example:

```
.netnetsystem.com
```

   The domain suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is ACME-abc123, the resulting FQDN is ACME-abc123.netnetsystem.com.

6. external-proxy-address—Enter the external proxy address to identify the IPv4 address of the SIP element (for example, a SIP proxy) in the external network with which the E-SBC communicates.

   Enter the value in the IP address format. For example:

```
192.168.11.200
```

7. external-proxy-port—Enter the external proxy port value to identify the UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the E-SBC communicates. The default is 5060. The valid range is:

   - Minimum—1025
   - Maximum—65535

8. external-address—Enter the external address, which is an IPv4 address on the media interface in the external realm.

   Enter the value in the IP address format. For example:

   ```
   192.168.11.101
   ```

   This value must be such that any packet with an external address value as its destination address is routed to the E-SBC through the media interface connected to or routable from the external realm.

9. home-address—Enter the home address, which is an IPv4 address on the network interface in the home realm. This value must be such that any packet with a home address value as its destination address must be routed to the E-SBC through the media interface connected to or routable from the home realm.

   Enter the value in the IP address format. For example:

   ```
   127.0.0.10
   ```

   The value entered in this field must be different from the IP address value of the home realm's network interface element.

   The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.

10. home-proxy-address—Enter the home proxy address to set the IP address for the home proxy (from the perspective of the external realm).

    By default, this field is empty. No home proxy address entry signifies there is no home proxy, and the external address will translate to the address of the E-SBC's SIP proxy.

    Enter the value in the IP address format. For example:

    ```
    127.1.0.10
    ```

11. home-proxy-port—Enter the home proxy port to set the port number for the home realm proxy. The default value is 0. The valid range is:

    - Minimum—0, 1025
    - Maximum—65535

12. route-home-proxy—Optional. Enable or disable requests being routed from a given SIP-NAT to the home proxy. The default value is disabled. The valid values are:

    - enabled—All inbound requests for a specific SIP NAT are routed to the home proxy
    - disabled—All inbound requests are not routed through the home proxy.
    - forced—The Request is forwarded to the home proxy without using a local policy.

13. address-prefix—Optional. Indicate the IPv4 address prefix from incoming messages that requires SIP NAT function encoding (regardless of the realm from which these messages came).

    Note: This value overrides the value set in the realm's address prefix field.

    This field's format incorporates an IPv4 address and number of bits in the network portion of the address. For example, a Class C address has a 24-bit network part. The address prefix for 101.102.103.x would be represented as 10.102.103.0/24.

    The default value is an asterisk (*). When you enter this value or do not enter a value, the realm's address prefix value is used.

14. tunnel-redirect—Set to one of the following values to indicate whether certain headers in a 3xx Response message received by the E-SBC are NATed when sent to the initiator of the SIP INVITE message. The default is disabled. The valid values are:

    - enabled—Certain headers in a 3xx Response message are NATed.
    - disabled—Certain headers in a 3xx Response message are not NATed.

15. use-url-parameter—Establish whether SIP headers will use the URL parameter (configured in the next step) for encoded addresses created by the SIP NAT function. If SIP headers will be used, this value identifies which types of headers will use the URL parameter. The default value is none. The available values include:

    - none—No headers will use the URL parameter for address encoding.

      The following example illustrates the functionality of an E-SBC using a use url parameter value of none:

      ```
      sip: 1234@1.2.3.4 is translated into sip: 1234-acme-xxxx@5.6.7.8
      ```

      where -acme-xxxx is a cookie and xxxx is the encoded version of 1.2.3.4.
    - from-to—From and To headers will use the URL parameter for address encoding

      The following example illustrates the functionality of a E-SBC using a use url parameter value of none:

      ```
      sip: 1234@1.2.3.4 is translated into sip: 1234@5.6.7.8; pn=acme-xxxx
      ```

      where -acme-xxxx is a cookie and xxxx is the encoded version of 1.2.3.4.
    - all—All headers will use the URL parameter for address encoding. Acme Packet recommends not using this values because other SIP elements or implementations (other than the Oracle Enterprise Session Border Controller) might not retain the URL parameter in subsequent SIP messages that they send to the Oracle Enterprise Session Border Controller.
    - phone—

      If this field is set to either from-to or all, the E-SBC puts the encoded address of the SIP NAT into a URL parameter instead of using the encoding name inside the userinfo part of the address.

16. parameter-name—If you have configured the use-url-parameter with the from-to or all value, you need to indicate the hostname prefix.

    The parameter name value is used in SIP NAT encoding addresses that have the use url parameter values of from-to or all.

17. user-NAT-tag—Enter a value to identify the username prefix used for SIP URIs. The values you can use can include any characters valid for the userinfo part of a URI. This should be made unique for each realm and SIP NAT function.

    The default value is -acme-.

    In combination with the domain suffix and host NAT tag values, this value is used to help the E-SBC identify an encoded URI that it needs to translate when moving between public and private realms.

18. host-NAT-tag—Enter a value for the host NAT tag field to identify the hostname prefix used for SIP URIs. The value refers to domain labels and can include any characters valid for the hostname part of the URI. This should be made unique for each realm and SIP NAT function.

    The default value is ACME-.

    In combination with the domain suffix and user NAT tag values, this value is used to help the E-SBC identify an encoded URI that it needs to translate when moving between public and private realms.

19. headers—List the SIP headers you want affected by the SIP NAT function. The URIs in these headers are translated and encrypted, and encryption occurs according to the SIP NAT function rules.

    To enter the full default list, type headers, followed by a Space and -d, then press Enter.

    You can also insert the following tags in SIP NAT headers if you want to replace FQDNs with next hop or SIP interface IP addresses:

    - fqdn-ip-tgt: replaces the FQDN with the target address
    - fqdn-ip-ext: replaces the FQDN with the SIP NAT external address

      Enter the tag using the following format:

      <header-name>=<tag>

      For example:

      To=fqdn-ip-tgt

The FQDN in a To header is replaced with the target IP address.

You can insert the following tags to apply NAT treatment to a From header in an INVITE when the gateway sends it into the home realm.

- ip-ip-tgt: replaces any IP address in the From header with the next hop target
- ip-ip-ext: replaces any IP address in the From header with the E-SBC's external address

To view all SIP NAT function parameters, enter a ? at the system prompt. The following example shows SIP NAT configuration for peering network.

```
sip-nat
        realm-id                        peer-1
        domain-suffix                   .p1.acme.com
        ext-proxy-address               192.168.11.200
        ext-proxy-port                  5060
        ext-address                     192.168.11.101
        home-address                    127.0.0.10
        home-proxy-address              127.1.0.10
        home-proxy-port                 5060
        route-home-proxy                enabled
        address-prefix                  *
        tunnel-redirect                 disabled
        use-url-parameter               none
        parameter-name
        user-nat-tag                    -p1-
        host-nat-tag                    P1-
        headers                         Call-ID Contact From Join Record-
Route
                                        Refer-To Replaces Reply-To Route
To Via
                                        f i m r t v
```

SIP Realm Bridging

This section explains how to configure the internal routing among realms known as realm bridging. Realm bridging lets you cross-connect SIP interfaces. You can use one of the following two methods for bridging realms:

- local policy bridging: use this method to enable dynamic internal routing between realms if your SIP interfaces do not have the SIP NAT function applied.
- SIP NAT bridging: use this method if your SIP interfaces have the SIP NAT function applied.

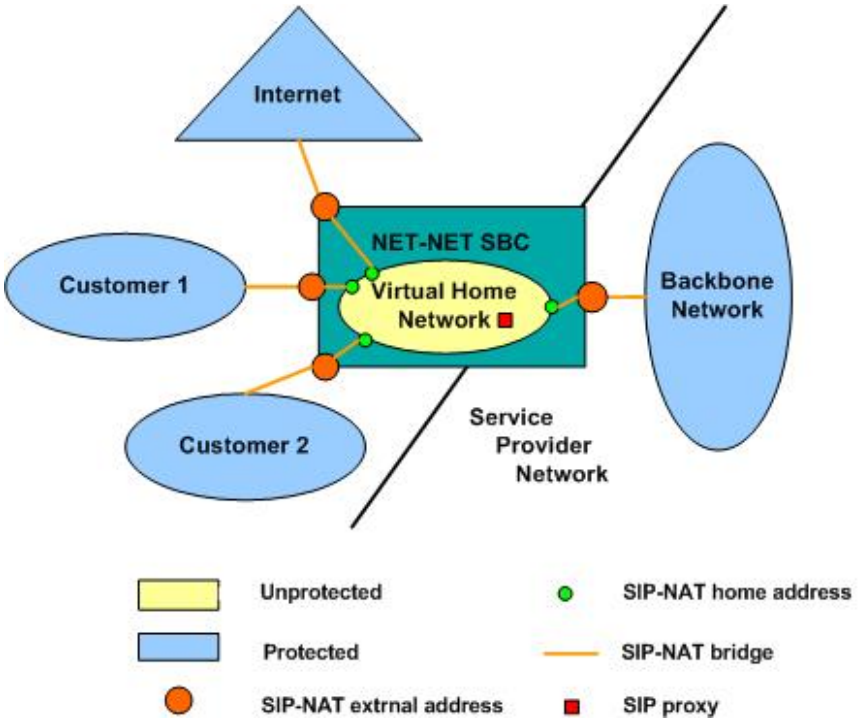
About SIP NAT Bridging

Each SIP NAT has a presence in two realms, trusted and untrusted. The SIP NAT bridge is the conduit for packages in and out of the home realm. It creates a bridge between realms by providing address translations; removing all references to the original IP addressing from the packets sent to the destination network.

With the SIP NAT bridge, an untrusted (or public) home network can reside within the Oracle Enterprise Session Border Controller, while the other entities (the backbone network, the Internet, or customer networks) are all trusted (or private). One of the primary functions of the SIP NAT bridge is to protect networks from one another so that address bases can remain hidden. Using a SIP NAT bridge, no one network has direct access to the data of other networks.

Establishing a SIP NAT bridge lets you route every SIP Request message through the backbone. Without using this functionality, it would appear as though all messages/sessions were coming from the Oracle Enterprise Session Border Controller's SIP proxy (the SIP server that receives SIP requests and forwards them on behalf of the requestor).

The following diagram illustrates this unprotected (or public) and protected (or private) division.



SIP NAT Bridge Configuration Scenarios

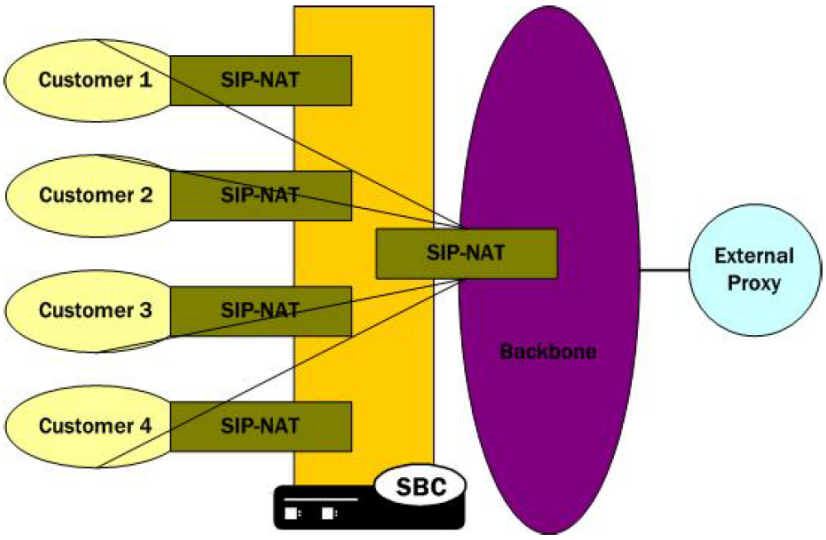
You can configure the SIP NAT bridge functionality in a many-to-one or a one-to-one relationship. For example, multiple customer SIP NATs can be tied to a single backbone SIP NAT, or a single customer SIP NAT can be tied to a single backbone SIP NAT.

You might need to use several SIP NATs on the customer side while using only one on the backbone side in a many-to-one relationship. Or you might configure one SIP NAT on the backbone side for every one that you configure on the customer side in a one-to-one relationship.

You can route all customer side SIP NAT requests to the corresponding backbone SIP NAT regardless of the Request URI. If a request arrives from the customer network with a Request URI that does not match the customer SIP NAT external address or the local policy that would route it to the backbone SIP NAT; the route home proxy value is used.

Many to One Configuration

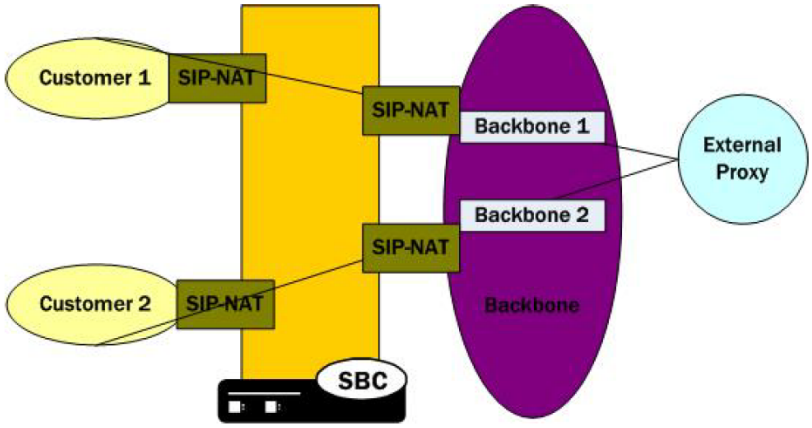
In the many-to-one scenario, multiple customer SIP NATs are tied to a single backbone SIP NAT. The following diagram illustrates the many-to-one SIP NAT bridge configuration.



One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. On the backbone SIP NAT side, you configure the home proxy address to match the home address of the customer SIP NAT. On the customer side, you configure the home proxy address to match the home address of the backbone SIP NAT.

The following diagram illustrates the one-to-one SIP-NAT bridge configuration.



SIP NAT Bridge Configuration

You create a bridge between SIP NATs by pointing them at one another. You point the SIP NATs at each other by configuring the home address and home proxy address to create the bridge. In addition, you can configure the route home proxy on the customer's side of a SIP NAT to force all requests to be routed to the corresponding backbone SIP NAT, regardless of the Request URI. You need to force requests when elements in the customer's network send requests with a Request URI that does not match the customer's SIP NAT external address. Or when the Request URI does not match a local policy element that would route the requests to the backbone SIP NAT.

You also need a home network to create a SIP NAT bridge. If you do not have a real home network, you need to create a virtual one. You also need to configure instances of the SIP NAT to create the SIP NAT bridge within your network.

Creating a Virtual Home Network

A virtual home network is a home network that resides entirely within the Oracle Enterprise Session Border Controller, as does a real home network. The difference between the two is the real home network also has a physical connection to the Oracle Enterprise Session Border Controller.

The internal home realm/network is usually configured with addresses within the special loopback range (127.0.0.0/8) as described in RFC 3330. This applies to the SIP port addresses for the home realm's SIP interface, and all home addresses for SIP NATs. The address 127.0.0.1 should not be used because it conflicts with the default loopback interface setup by the system for inter-process communication.

To create a virtual home network:

1. Set the name and subport ID of the network interface associated with the home realm element to lo0:0.
2. To enable the SIP proxy to listen for messages on the virtual home realm, configure the home realm ID. It must correspond to the realm's identifier, in which you set the network interface subelement to point to the appropriate network interface element.

   The following table lists the field values you need to set when you are using SIP NAT bridge functionality and you do not have a real home network.

Configuration Element		Sample Values
realm configuration	identifier	home
	network interfaces	lo0:0
	address prefix	127.0.0.0/8
SIP configuration	home realm ID	home
	SIP ports address	127.0.0.100

Many-to-One Configuration

To configure many-to-one:

1. For the backbone SIP NAT, ensure the home proxy address field is blank.
2. For the customer side SIP NAT:

   Set the home address to match the home address of the customer.

   Set the home proxy address to match the backbone SIP NAT home address.

   Set route home proxy to forced.

   The following table lists the field values you need to set to create a many-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IPv4 address of the home realm. For example: 127.0.0.120
	home proxy address	IPv4 address of the home proxy from the perspective of the external realm. For a backbone SIP NAT, leave blank.
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110
	route home proxy	forced

One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. The home proxy address field value of the backbone SIP NAT must match the home address of the customer SIP NAT. On the customer side, the home address of the customer SIP NAT should be defined as the home address of the customer, the home proxy address field value should match the home address of the backbone SIP NAT, and route home proxy should be set to forced.

The following table lists the field values you need to set to create a one-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IPv4 address of the home realm. For example: 127.0.0.110
	home proxy address	IPv4 address of the home proxy from the perspective of the external realm. 127.0.0.120
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110
	route home proxy	forced

Shared Session Agent

Usually, the same set of servers (the external proxy) is used for all SIP NATs to the backbone network. In order to support redundant servers in the backbone of a SIP NAT bridge, the original egress realm as determined by the incoming Request URI needs to be retained after a local policy lookup.

When a request arrives at the Oracle Enterprise Session Border Controller, it determines the matching (target) session agent and, after the local policy is examined, sets the new outbound session agent to the one from the selected target.

If the target session agent's realm is set to *, the Oracle Enterprise Session Border Controller retains the original session agent's realm ID. Because the target session agent does not have a realm ID defined, the original egress realm is retained.

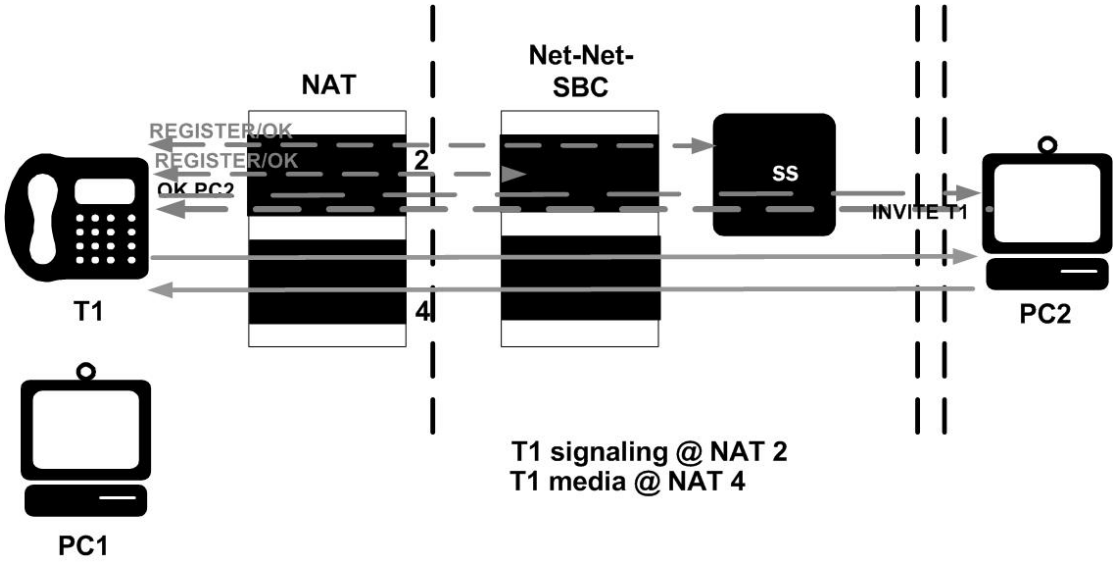
SIP Hosted NAT Traversal (HNT)

This section explains how to configure SIP Hosted Network Address Translation (HNT) traversal. SIP HNT lets endpoints behind a NAT/firewall device send and receive signaling and media using the Oracle Enterprise Session Border Controller as a relay.

About SIP HNT

SIP HNT is a technique the Oracle Enterprise Session Border Controller uses to provide persistent reachability for SIP UAs located in private Local Area Networks (LANs) behind Nat/firewall devices. It relies on frequent, persistent messaging to ensure that the binding on the intermediary NAT device is not torn down because of inactivity. HNT does not require support for the NAT in the SIP endpoint.

The following diagram illustrates SIP HNT traversal.z



The Oracle Enterprise Session Border Controller's HNT function allows endpoints located behind NATs to communicate; providing means to traverse NATs. The Oracle Enterprise Session Border Controller interacts with endpoints (using SIP) to allow persistent inbound and outbound signaling and media communications through these NATs.

The Oracle Enterprise Session Border Controller automatically detects when an intermediate NAT exists between the UA and the Oracle Enterprise Session Border Controller by comparing the Layer 3 IP address of a REGISTER message with the IP address indicated within the UA. The Oracle Enterprise Session Border Controller sends signaling responses to the address and port that the request came from, rather than the address and port indicated in the request. The Via header in the request message indicates where the response should be sent.

Using HNT with Existing NAT Device

For network architectures in which premise devices and endpoints reside behind an existing NAT device, the Oracle Enterprise Session Border Controller's HNT function allows these premise NATs to be traversed without requiring an upgrade to the premise equipment, the deployment and management of additional premise-based hardware or software, or any NAT device configuration changes.

Registering Endpoints

The Oracle Enterprise Session Border Controller uses periodic endpoint registration messages to dynamically establish and maintain bindings in the NAT. These bindings keep a signaling port (port that is opened on a firewall to allow traffic to pass through it is a pinhole) open in the NAT that allows the inbound signaled communications to pass through. Using the endpoint registrations, the Oracle Enterprise Session Border Controller then maps the Layer 3 (OSI network layer that deals with switching and routing technologies for data transmission between network devices) IPv4 address/port information from the NAT device to the Layer 5 (OSI session layer that deals with session and connection coordination between applications) entity (for example, user name or phone number) behind the NAT so that when an incoming signaling message is received, the Oracle Enterprise Session Border Controller sends it to the appropriate address and port on the NAT for the called party.

Establishing Media Flows

During call setup, the ports for bidirectional media flows are established dynamically. Since the media flows also pass through the Oracle Enterprise Session Border Controller, it can identify the IPv4 address/port information on the NAT device used for the outgoing media coming from the user name/phone number. The Oracle Enterprise Session Border Controller then uses that same NAT's IPv4 address/port information to send incoming media to the correct user name/phone number behind the NAT device.

Prerequisites

In order to achieve HNT, the endpoints involved must be capable of:

- symmetric signaling: sending and receiving SIP messages from the same transport address (IP address or User Datagram Protocol/Transmission Control Protocol (UDP/TCP) port
- symmetric media: sending and receiving Real-Time Transport Protocol (RTP) messages from the same UDP port

These conditions are required to allow signaling and media packets back through the NAT (through the bound external address and port). These packets must come from the address and port to which the outbound packet that created the NAT binding was sent. The NAT sends these inbound packets to the source address and port of the original outbound packet.

When SIP HNT is used, the Oracle Enterprise Session Border Controller sends signaling responses to the address and port that the request came from rather than the address and port indicated in the request. The Via header in the request message indicates where the response should be sent.

Keeping the NAT Binding Open

Additional measures are also required to keep the NAT binding open because most NAT bindings are discarded after approximately a minute of inactivity. The Oracle Enterprise Session Border Controller keeps the SIP NAT binding open by returning a short expiration time in REGISTER responses that forces the endpoint to send frequent REGISTER requests.

In order to keep the NAT binding open for SIP, the Oracle Enterprise Session Border Controller maintains the registration state. When an endpoint first registers, the Oracle Enterprise Session Border Controller forwards that REGISTER message on to the real registrar. You can define the real registrar using either of the following methods:

- Configure the SIP config registrar host and registrar port to indicate the real registrar.
- Map the SIP config registrar host and registrar port values to the SIP NAT home proxy address and home proxy port values. Then configure the SIP NAT's external proxy address and external proxy port values to correspond to the real registrar.

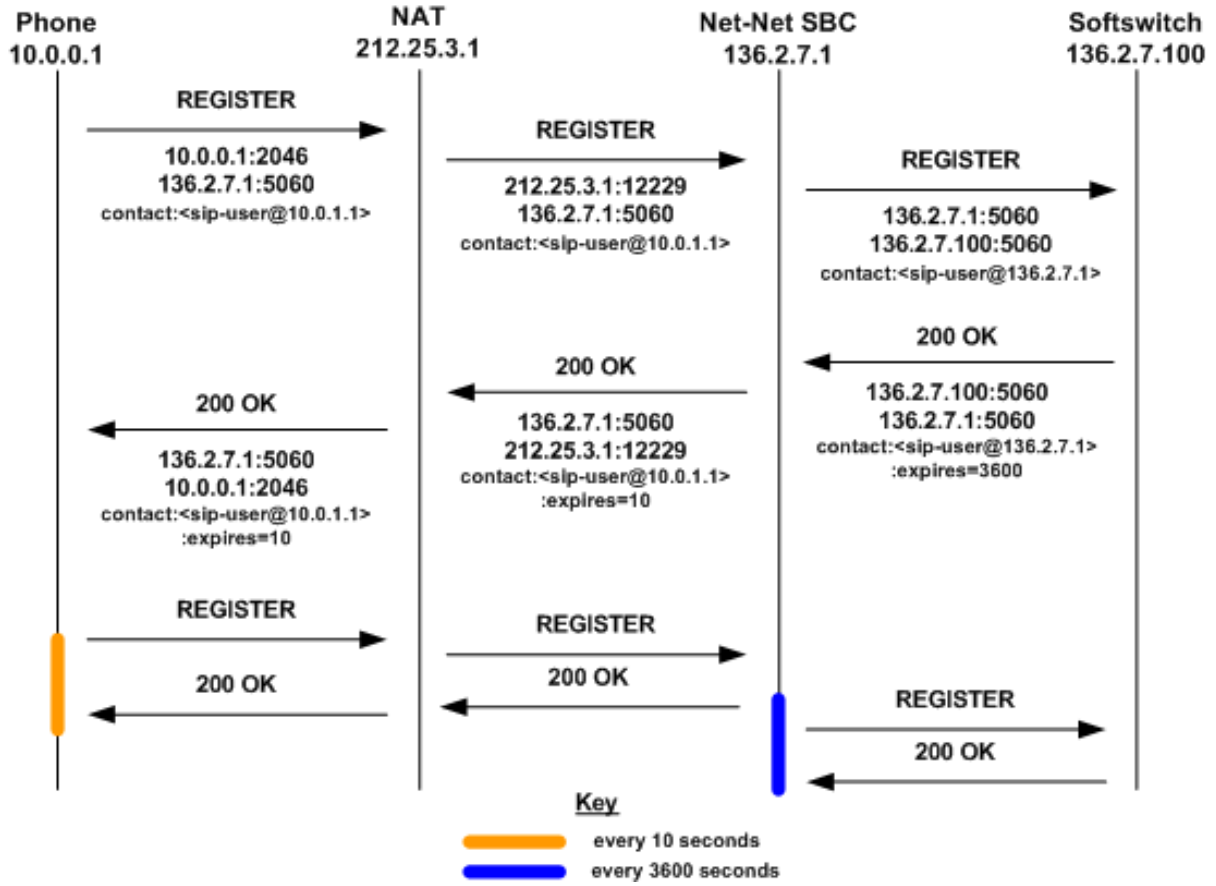
Note: A registrar can be located in a SIP NAT realm.

When a successful response is received, the Oracle Enterprise Session Border Controller caches the registration to memory. This cached registration lives for the length of time indicated by the expiration period defined in the REGISTER response message from the registrar. The response sent back to the endpoint has a shorter expiration time (defined by the SIP config's NAT interval) that causes the endpoint to send another REGISTER message within that interval. If the endpoint sends another REGISTER message before the cached registration expires, the Oracle Enterprise Session Border Controller responds directly to the endpoint. It does not forward the message to the real registrar.

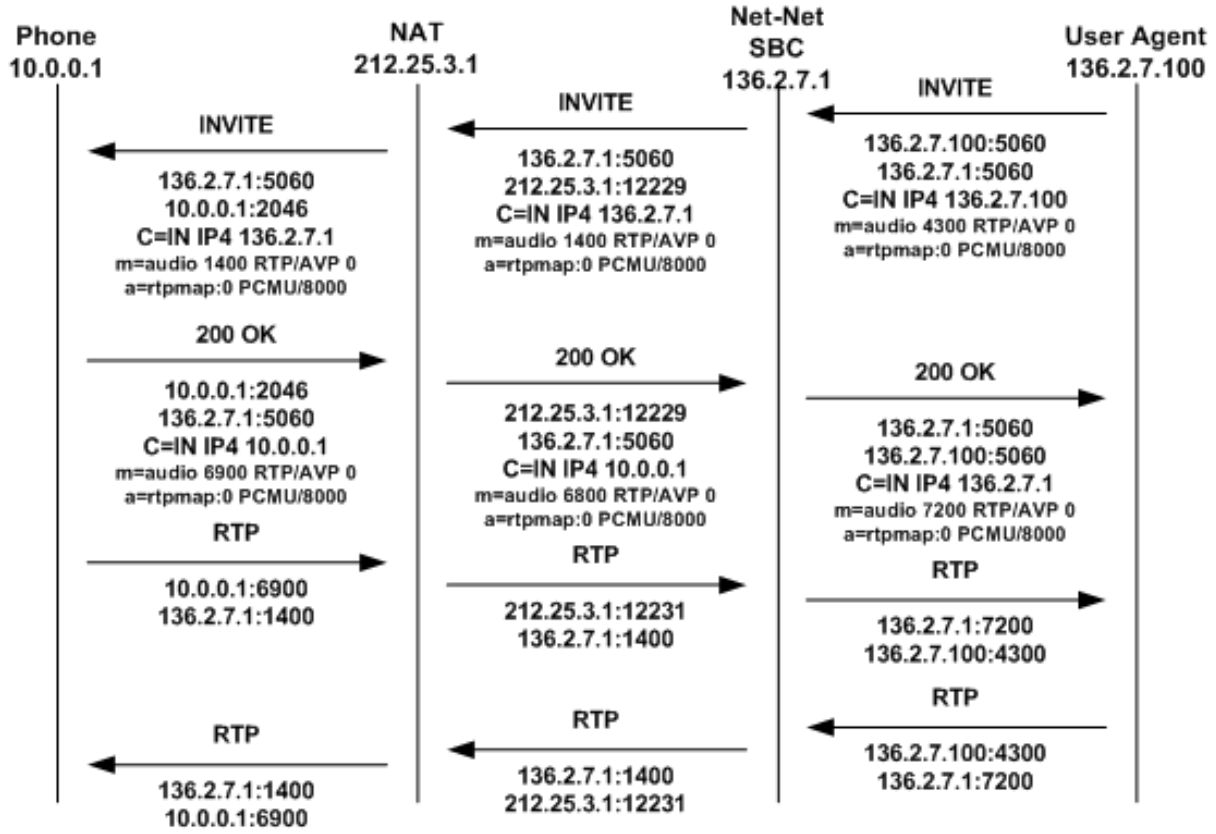
If the cached registration expires within the length of time indicated by the NAT interval, the REGISTER message is forwarded to the real registrar. If the Oracle Enterprise Session Border Controller does not receive another REGISTER message from the endpoint within the length of time indicated by the NAT interval, it discards the cached registration.

The Contact Uniform Resource Identifier (URI) in the REGISTER message sent to the registrar by the Oracle Enterprise Session Border Controller points at the Oracle Enterprise Session Border Controller so that the proxy associated with the real registrar sends inbound requests to the Oracle Enterprise Session Border Controller. This way, the inbound requests can be forwarded to the endpoint through the NAT binding.

The following example illustrates the SIP HNT registration call flow for the SIP HNT feature.



The following example illustrates the SIP HNT invitation call flow for the SIP HNT feature.



Working with Multiple Domains

You can use a wildcard (*) with the HNT feature to accommodate multiple domains and to allow the Oracle Enterprise Session Border Controller to cache all HNT endpoints. The wildcard functionality is enabled in the SIP config by entering an asterisk (*) in the registrar domain and registrar host fields.

The wildcard allows the use of either a local policy or Domain Name Service (DNS) to resolve the domain name to the correct registrar. Either method can be used to route the Fully Qualified Domain Name (FQDN) when the you enter an asterisk (*) for the register host. An FQDN consists of an unlimited number of domain labels (domain names), each separated by a dot (.). The FQDN can include the top level domain name (for example, acmepacket.com).

In the hostname acme-packet.domainlbl.example100.com, the syntax is as follows:

- acme-packet is a domain label
- domainlbl is a domain label
- example100 is a domain label
- com is the top label

The information configured in a local policy is used before DNS is used. If the next hop destination address (defined in the local policy's next hop field) is an IPv4 address, a DNS server is not needed. A DNS server is needed when the IPv4 address of the next hop destination address is a FQDN or cannot be determined from the Oracle Enterprise Session Border Controller's configuration. Even with a configured local policy, the next hop destination address might be an FQDN that requires a DNS lookup.

If the registrar host does not use the wildcard, the Oracle Enterprise Session Border Controller always uses the configured address. You can limit the number of endpoints that receive the HNT function. For example, you can use a non-wildcarded registrar domain field value (like acme.com) with a wildcarded registrar host field value.

HNT Configuration Overview

To configure SIP HNT NAT traversal, you need to configure both the SIP interface and the SIP config.

SIP HNT Single Domain Example

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a single domain and registrar.

- SIP config

Parameter	Sample Value
registrar domain	netnetsystem.com
registrar host	192.168.12.1
registrar port	5060

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60
minimum registration expire	200
registration caching	disabled
route to registrar	enabled

SIP HNT Multiple Domain Example

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a multiple domains and multiple registrars.

- SIP config

Parameter	Sample Value
registrar domain	*
registrar host	*
registrar port	0

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60
minimum registration expire	200
registration caching	disabled
route to registrar	enabled

HNT Configuration

To configure a SIP interface on the Oracle Enterprise Session Border Controller (E-SBC):

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

   From this point, you can configure physical interface parameters. To view all SIP interface parameters, enter a ? at the system prompt.

4. nat-traversal—Define the type of HNT enabled for SIP. The default value is none. Available values include:
   - none—Disables the HNT feature for SIP (default value)
   - rport—SIP HNT function only applies to endpoints that include the rport parameter in the Via header and the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address.
   - always—SIP HNT applies to requests when the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address. (Even when the rport parameter is not present.)
5. nat-interval—Set the expiration time in seconds for the E-SBC's cached registration entry for an HNT endpoint. The default value is 30. The valid range is:
   - Minimum—0
   - Maximum—999999999

Acme Packet recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to prevent the NAT device from keeping an unused port open.

6. registration-caching—Enable for use with all UAs, not just those that are behind NATs. By default, this field is set to disabled. If enabled, the E-SBC caches the Contact header in the UA's REGISTER request when it is addressed to one of the following:

   - E-SBC
   - registrar domain value
   - registrar host value

   The E-SBC then generates a Contact header with the E-SBC's address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.

   Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:

   - enabled—The E-SBC takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.
   - disabled—The user part of the Contact header is taken from the URI in the From header and no cookie is appended. This is the default behavior of the Oracle Enterprise Session Border Controller.

   When the registrar receives a request that matches the address-of-record (the To header in the REGISTER message), it sends the matching request to the E-SBC, which is the Contact address. Then, the v forwards the request to the Contact-URI it cached from the original REGISTER message.

7. min-reg-expire—Set the time in seconds for the SIP interface. The value you enter here sets the minimum registration expiration time in seconds for HNT registration caching. The default value is 300. The valid range is:

   - Minimum—1
   - Maximum—999999999

   This value defines the minimum expiration value the E-SBC places in each REGISTER message it sends to the real registrar. In HNT, the E-SBC caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

   Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the E-SBC to send frequent registrations on to the real registrar.

8. registration-interval—Set the E-SBC's cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the E-SBC to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires. The default value is 3600. The valid range is:

   - Minimum—1

   A registration interval of zero causes the E-SBC to pass back the expiration time set by and returned in the registration response from the registrar.

   - Maximum—999999999

   If the expiration time you set is less than the expiration time set by and returned from the real registrar, the E-SBC responds to the refresh request directly rather than forwarding it to the registrar.

   Note: With registration caching, there is no NAT; therefore, a short registration interval causes the UA to send excess REGISTER messages.

   Although the registration interval applies to non-HNT registration cache entries, and the loosely related NAT interval applies to HNT registration cache entries, you can use the two in combination. Using a combination of the two means you can implement HNT and non-HNT architectures on the same E-SBC. You can then define a longer interval time in the registration interval field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

9. route-to-registrar—Enable routing to the registrar to send all requests that match a cached registration to the destination defined for the registrar host; used when the Request-URI matches the registrar host value or the registrar domain value, not the SBC's address. Because the registrar host is the real registrar, it should send the requests back to the E-SBC with the E-SBC's address in the Request-URI. The default value is disabled. The valid values are:

   - enabled | disabled

     For example, you should enable routing to the registrar if your network uses a E-SBC and needs requests to go through its service proxy, which is defined in the registrar host field.

Global SIP Configuration

To configure the SIP configuration:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.

4. registrar-domain—Optional. Define the domain to match against the host part of a URI to determine if a request is addressed to the registrar. If there is a match, the registration caching, NAT traversal, and route to registrar parameter values for the SIP interface are applied to the request. By default, this field remains empty. Available values are:

   - an asterisk (*) to specify the values apply to all requests.
   - any alphanumeric character or any combination of alphanumeric characters. For example, acme1.com.

     A hostname consists of any number of domain labels, separated by dots (.), and one top label. A top label is the last segment of the hostname. It must start with an alphabetical character. After the first character, a top label can consist of any number or combination of alphanumeric characters, including those separated by dashes. The dash must be preceded and followed by alphanumeric characters. A single alphabetical character is the minimum requirement for a hostname field (for example, c to indicate .com).

     When the REGISTER message's Request-URI has an FQDN, it is matched against the registrar domain's value to determine if the message needs to be forwarded to the registrar port on the registrar host. The registrar domain's value is also used when route to registrar is set to enabled, to determine if a request needs to be forwarded to the registrar.

     Only the right-hand part of the domain name in the Request-URI needs to match the registrar domain value. For example, acme3.acmepacket.com matches acmepacket.com. However, the entire domain label within the domain name must match. For example, the domain label "acme3.acmepacket.com" would not match packet.com.

5. registrar-host—Define the address of the registrar for which requests for registration caching, NAT traversal, and router to registrar options apply. You can use a specific hostname, a IP address, or a wildcard (*):

   - an asterisk (*) indicates normal routing (local policy, DNS resolution, and so on) is used to determine the registrar's address.
   - hostname: can consist of any alphanumeric character or any combination of alphanumeric characters (for example, acme1.com). The hostname can consist of any number of domain labels, separated by dots (.), and one top label. You can use the minimum field value of a single alphabetical character to indicate the top label value (for example, c to indicate .com).

- IPv4 address: must follow the dotted notation format. Each of the four segments can contain a numerical value between zero (0) and 255. For example, 192.168.201.2. An example of a invalid segment value is 256.

  By default, the registrar host field remains empty.

6. registrar-port—Set the SIP registrar port number. The SIP registrar server configured in this and the registrar host field is the real registrar. Or the values entered in those fields map to the home proxy address and home proxy port of the SIP NAT with external proxy address and external proxy port values that correspond to the real registrar. The default value is 0. The valid range is:

   - Minimum—0, 1025
   - Maximum—65535

   The following example shows the values for a single domain and registrar configuration.

```
sip-config
        state                           enabled
        operation-mode                  dialog
dialog-transparency        disabled
        home-realm-id                   acme
        egress-realm-id
        nat-mode                        Public
        registrar-domain
        registrar-host
        registrar-port                  0
        init-timer                      500
        max-timer                       4000
        trans-expire                    32
        invite-expire                   180
        inactive-dynamic-conn           32
        red-sip-port                    1988
        red-max-trans                   10000
        red-sync-start-time             5000
        red-sync-comp-time              1000
        last-modified-date              2005-03-19 12:41:28
```

Keep-Alive with CR LF 2832

Release S-CX6.3F1 provides an alternative NAT (Network Address Translator) Traversal method. The current method is SBC-based and requires no explicit participation by the SIP endpoint. Rather the SBC manipulates SIP registration requests and responses to the endpoint — causing it to issue frequent and extraneous registration requests thus maintaining existing NAT bindings.

The alternative method is based upon RFC 5626, *Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)*, and RFC 6223. Unlike the current SBC-centric method, the new alternative requires the active participation of the SIP endpoint. With this method the SIP endpoint and the SBC negotiate a request / response message sequence which generates sufficient traffic flow to maintain NAT bindings.

Section 3.5.2 of RFC 5626 defines a keep-alive method for connectionless UDP flows, but provides no guidance for keep-alive negotiation. The Indication of Support for Keep-Alive internet draft addresses this deficiency by defining a procedure that enables a SIP endpoint to signal its capability and willingness to send and receive periodic keep-alive messages to a device referred to by the RFC as an edge proxy, a role performed by the SBC. After receiving such a signal, the SBC returns a response indicating its willingness to exchange keep-alives, and specifying the frequency of the exchange.

SIP endpoints that initiate and participate in the keep-alive exchanges described in this section must support a minimal sub-set of client operations. Specifically, endpoints must be able to construct and transmit CR/LF binding requests, and receive and parse CR/LF binding responses.Binding request and response formats are described in Section 6 of RFC 5626.

As shown in the following SIP Registration request, the SIP endpoint, functioning as a CR/LF client, signals its willingness to exchange keep-alive messages by placing an unvalued keep parameter, newly-defined by the [RFC]

6223 in the SIP Via header. The expires parameter in the Contact header requests a registration period of 5 hours (18000 seconds).

```
REGISTER sip:512@172.16.101.23:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.101.38:5070;branch=dd1;keep
From: "512" <sip:512@172.16.101.38:5070;transport=UDP>;tag=443322
To: "512" <sip:512@172.16.101.38:5070>
Call-ID:1-14400@172.16.101.38
CSeq: 1 REGISTER
Max-Forwards: 70
User-Agent: ADTRAN_Total_Access_908e_(2nd_Gen)/A1.02.00.E
Content-Length: 0
```

The SBC forwards the Registration request (absent the keep parameter) to the Registrar.

```
REGISTER sip:512@192.168.7.32:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.101.23:5060;branch=z9hG4bK3q43klh3dafekdepnbaqip04e1
From: "512" <sip:512@172.16.101.38:5070;transport=UDP>;tag=443322
To: "512" <sip:512@172.16.101.38:5070>
Call-ID: 1-14400@172.16.101.38
CSeq: 1 REGISTER
Max-Forwards: 69
Contact: "512" <sip:512@192.168.101.23:5060;transport=udp>;expires:18000
User-Agent: ADTRAN_Total_Access_908e_(2nd_Gen)/A1.02.00.E
Content-Length: 0
Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, REFER
```

The Registrar indicates successful registration with a 200 OK response back to the SBC. The expires parameter in the Contact header grants a registration period of 1 hour (3600 seconds).

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.101.23:5060;branch=z9hG4bK3q43klh3dafekdepnbaqip04e1
From: "512" <sip:512@172.16.101.38:5070;transport=UDP>;tag=443322
To: "512" <sip:512@172.16.101.38:5070>;tag=b68b3d53a5a90225609112ff6c211bef.
16a6
Call-ID: 1-14400@172.16.101.38
CSeq: 1 REGISTER
Contact: <sip:512@192.168.101.23:5060;transport=udp>;expires=3600
Server: OpenSER (1.3.0-notls (i386/linux))
Content-Length: 0
```

The SBC, forwards the 200 OK to the endpoint after inserting a keep parameter and a parameter value in the Via header of the Registration response. The presence of the keep parameter signals the SBC's willingness to exchange keep-alives, and the parameter value specifies the exchange frequency in seconds.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.16.101.38:5070;branch=dd1;keep=20
From: "512" <sip:512@172.16.101.38:5070;transport=UDP>;tag=443322
To: "512" <sip:512@172.16.101.38:5070>;tag=b68b3d53a5a90225609112ff6c211bef.
16a6
Call-ID: 1-14400@172.16.101.38
CSeq: 1 REGISTER
Contact: <sip:512@172.16.101.38:5070>;expires=3600
Server: OpenSER (1.3.0-notls (i386/linux))
Content-Length: 0
```

After the keep-alive exchange has been negotiated, the SIP endpoint, acting as a CR/LF client, is required to transmit a periodic CR/LF so that the interval between each request is randomly distributed between 80 and 100 percent of the value of the keep parameter. Assuming a parameter value of 20 seconds, for example, the SIP endpoint transmits a CR/LF at random intervals between 16 and 20 seconds in length.

Upon receipt of a Ping, the SBC, transmits a Pong. Receipt of the Pong by the endpoint confirms the TCP connection between the endpoint and the SBC, and the viability of NAT bindings in the transmission path.

Once initiated, endpoint transmission of CR/LF Ping and SBC responses continue for the duration of the SIP Registration, 1 hour in the above example, or until the endpoint transmits a new Registration request. In the event of

such a request, the endpoint must once again indicate its willingness to exchange CR/LF keep-alives with an unvalued keep parameter in the Via header. If keep-alive renegotiation is not successful, the endpoint must cease the transmission of keep-alive messages.

An endpoint failure to issue a timely CR/LF Ping is not fatal. In the absence of an expected request, the SBC takes no action with regard to the TCP connection, or to established sessions.

Keep-alive Configuration

You use the register-keep-alive attribute, available in SIP Interface configuration mode, to enable CR/LF keep-alive on a SIP interface.

1. In Superuser mode, use the following ACLI command sequence to access SIP Interface configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

2. The register-keep-alive attribute enables CR/LF keep-alive on the current SIP interface.

   none — (the default) disables CR/LF keep-alive

   always — assuming that the endpoint has included the keep parameter in the Via header, exchange CR/LF keep-alives with that endpoint

   bnat — assuming that the endpoint has included the keep parameter in the Via header, exchange CR/LF keep-alives only with endpoints that are behind an intervening NAT device (based on comparing source IP packet addresses with IP addresses extracted from the SIP request).

```
ACMEPACKET(sip-interface)# register-keep-alive always
```

   ACMEPACKET(sip-interface)#

3. If CR/LF keep-alive is enabled on the current SIP interface (register-keep-alive is always or hint), use the tcp-nat-interval attribute to specify the value of the keep parameter provided by the SBC to the SIP endpoint.

   In the absence of an explicit assignment, this attribute defaults to a value of 30 seconds.

   The SIP endpoint transmits periodic CR/LF Ping so that the interval between each request is randomly distributed between 80 and 100 percent of the value of the tcp-nat-interval attribute.

   Assuming the default value (30 seconds) the interval between CR/LF binding requests would vary from 24 to 30 seconds.

```
ACMEPACKET(sip-interface)# nat-interval 20
```

   ACMEPACKET(sip-interface)#

4. Use done, exit, and verify-config to complete this configuration.
5. Save and activate your configuration.

SIP Registration Local Expiration

When you deploy multiple Oracle Enterprise Session Border Controllers (E-SBC) in series and they have registration caching and HNT configured, registration cache entries might expire prematurely in instances with several devices provisioned with the same address of record (AoR). Now you can configure a SIP interface option to prevent the premature expiration.

When you use registration caching and HNT, the E-SBC adjusts the expiration time it sends to user agents (UAs) in REGISTER responses based on the registration interval you configure. It can be the case that a SIP user has multiple registered contact endpoints at the UA to which a response is sent. If the URI in the Contact contains the UA‘s address and that UA included the Contact in the REGISTER request, then the Contact is seen as exclusively belonging to that UA. In the REGISTER response, this Contact (exclusive to the UA) includes the local expiration time, a time based on the SIP interface configuration’s registration or NAT interval value. Additional Contacts (not

exclusive to the UA) in the REGISTER response have the expiration time from the REGISTER response the registrar sent to the E-SBC.

It is this default behavior can cause registration cache entries to expire prematurely in the E-SBC nearest a registrar when multiple E-SBCs are deployed in series. Multiple registering UAs for a single SIP user, for example, might trigger the early expiration. The SIP you can configure an option per SIP interface that causes the E-SBC to send the local registration expiration time in all in the Expires parameter of all Contact headers included in REGISTER responses sent from the SIP interface.

SIP Registration Local Expiration Configuration

You can configure this feature either for the global SIP configuration, or for an individual SIP interface.

To configure SIP registration local expiration for the global SIP configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. If you are editing an existing configuration, select the configuration so you can enable this feature.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name reg-local-expires with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-config)# options +reg-local-expires
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

   To configure SIP registration local expiration for an individual SIP interface:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type sip-interface and press Enter. If you are editing an existing configuration, select the one on which you want to enable this feature.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

9. options—Set the options parameter by typing options, a Space, the option name reg-local-expires with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-interface)# options +reg-local-expires
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

10. Save and activate your configuration.

Simultaneous TCP Connection and Registration Cache Deletion

You can configure the Oracle Enterprise Session Border Controller to automatically start a timer when a user deregisters or changes location by registering with a new contact address.

Not all devices tear down TCP connections associated with these old addresses when a user registers with a new contact address.

Registration Cache Deletion Configuration

You can apply suppress-reinvite to the sip-interface facing the User Agents whose re-INVITEs are to be responded to locally.

To enable Registration Cache Deletion:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. tcp-conn-dereg—The delay (in seconds) that is used to determine when to terminate the TCP connection for a user that has been removed from the registration cache or changed location. This feature can be disable by setting the value to zero.

```
ACMEPACKET(session-agent)# tcp-conn-dereg 5300
```

5. Save your work.

SBC Incorrectly Appends Cookie in SIP REGISTER Message

The Oracle Enterprise Session Border Controller does not recognize a SIP URI containing tel-URI information if it doesn’t also contain a "user=phone" parameter. This behavior adversely affects creation of the acme_nat tag and placement of the cookie

You can enable the option proces-implicit-tel-URI to recognize an implicit tel-URI and places the cookie in the correct location.

process-implicit-tel-URI Configuration

To enable proccess-implicit-tel-URI

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option-name proccess-implicit-tel-URI with a plus sign in front of it, and then press Enter.

   ```
   ACMEPACKET(sip-config)# options +proccess-implicit-tel-URI
   ```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the SIP interface configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.
5. Save and activate your configuration.

SIP HNT Forced Unregistration

If you use HNT and experience the issue explained in this section, consider using the Oracle Enterprise Session Border Controller (E-SBC) forced unregistration feature. When this feature is enabled and a registration entry for an endpoint expires, the E-SBC notifies the soft switch to remove this binding using REGISTER message. In that REGISTER message, the expires header will be set to 0 and the expires parameter in the Contact header will also be set to 0.

The benefits of using forced unregistration include:

- Leveraging existing HNT configuration to provide near real-time information about the UA's status to the registrar/softswitch
- Preserving resource utilization for the E-SBC and the softswitch by deleting a contact binding that is no longer valid or needed
- Preventing extra bindings from being generated at the softswitch (e.g., in instances when the UA or NAT restart)

This feature applies to:

- HNT endpoints with registration caching enabled by default, and when the nat-traversal parameter in the SIP interface configuration is set to always
- non-HNT endpoints with registration caching enabled, when the registration-interval parameter in the SIP interface configuration is used in the expires header sent to the UA in the 200 OK

When to Use Forced Unregistration

For typical HNT use, it is common that the registration interval between the client UA and the Oracle Enterprise Session Border Controller (E-SBC) is between 60 and 120 seconds. This differs significantly from the re-registration interval between the E-SBC and the and the registrar, which varies from approximately 30 to 60 minutes.

If the UA fails to refresh its registration, the contact binding at the E-SBC is deleted after the registration expires. This expiration is determined by the expires= header in the 200 OK. The binding at the real registrar will remain intact. This creates a discrepancy between the real state of the UA and state of the softswitch. In the best case scenario, the contact binding expires at the softswitch after a few minutes.

For network management, this discrepancy can be problematic because the service provider would be unaware of the UA's status until the binding expires at the softswitch. This can take a considerable amount of time to happen.

In addition, the E-SBC encodes a cookie in the userinfo of the Contact header in the REGISTER message. This is a function of the source IPv4 address and port from which the request came, i.e., the ephemeral port in the NAT for DSL scenarios. Therefore, additional bindings that remain for long periods of time are created at the registrar if, for example, the:

- UA reboots
- Ethernet link between the UA and the DSL router is lost for over two minutes
- DSL crashes
- DSL/ATM layer between the DSL router

Caution for Using Forced Unregistration

You should use caution when applying SIP HNT forced unregistration for the following reasons:

- It can have an impact on the performance of your Oracle Enterprise Session Border Controller and the registrar, especially when you have a large number of HNT endpoints in your configuration that become unavailable simultaneously.
- It is possible that the registrar might become vulnerable to overload in the case where the registrar must authenticate a large number of register messages generated when HNT endpoints are de-registered. It is possible that the cached registration credentials might become "stale" over time (e.g., the nonce value usually has a limited lifetime). Without proper credentials, the registrar will reject the de-registrations.

Given these concerns, we recommend that you consult with your Oracle systems engineer before adopting the use of forced unregistration.

SIP HNT Forced Unregistration Configuration

To enable SIP HNT forced unregistration:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ```

4. Use the ACLI select command so that you can work with the SIP configuration.

   ```
   ACMEPACKET(sip-config)# select
   ```

5. options—Set the options parameter by typing options, a Space, the option name force-unregistration, and then press Enter.

   ```
   ACMEPACKET(sip-config)# options +force-unregistration
   ```

   If you type options force-unregistration, you will overwrite any previously configured options. In order to append the new option to the sip-config's options list, you must prepend the new option with a plus sign as shown in the previous example.

Adaptive HNT

This section explains how to configure adaptive HNT. The adaptive HNT expires feature allows the Oracle Enterprise Session Border Controller to automatically determine the maximum SIP REGISTER message expires time interval in order to keep each individual NAT pinhole open when performing SIP HNT. This feature applies only to SIP over UDP.

Overview

Without adaptive HNT, the Oracle Enterprise Session Border Controller keeps NAT pinholes open and port mapping cached by forcing the UAC to send frequent SIP REGISTER messages. It does so by setting the expires time to a short interval. Some NATs only need a message to be sent by the private client once every twenty minutes, while other NATs delete their cache/pinhole in thirty seconds if no messages appear. Given this large variation in time intervals, the Oracle Enterprise Session Border Controller's nat-interval (expire time) has been set to a low value in order to support as many NAT types as possible. However, CPU performance and scalability issues result from such a small refresh time, especially when there is a very large number of potential registered users.

When you use adaptive HNT, the Oracle Enterprise Session Border Controller waits for a time interval and then sends a SIP OPTIONS message to the UAC to see if it can still be reached. If the UAC can still be reached, the Oracle Enterprise Session Border Controller increases the timer and tries again. In case the pinhole closes because it has exceeded the NAT's cache time, the Oracle Enterprise Session Border Controller sets the expires time to be slightly longer than the time it tests using the OPTIONS method. This way, the UAC will send another REGISTER message shortly thereafter and impact on service will be minimal.

Adaptive HNT Example

An example call flow using adaptive HNT involves a basic HNT user and a Oracle Enterprise Session Border Controller. It begins when the Oracle Enterprise Session Border Controller receives and forwards the 200 OK for the REGISTER message. Then the Oracle Enterprise Session Border Controller sends an expires timer for slightly longer than the time for which to test; in this example, it begins the test for the amount of time set for the minimum NAT interval. It adds ten seconds to this time when it sends the expires timer. This way, there is time for the OPTIONS message to be sent before the REGISTER message is received (which would refresh the NAT's cache). The Oracle Enterprise Session Border Controller also tries to keep the REGISTER time short enough so that even if the NAT pinhole closes, there is minimal time before the UAC creates a new NAT binding by sending another REGISTER. Because a ten second interval may be too long, you might want to set this value to a better-suited time.

The test succeeds with a minimum test-timer because the UAC responded to the OPTIONS message. So the test-timer value is increased by thirty seconds and tried again. The expires time in the REGISTER message will be increased to the test-timer value plus ten seconds. This time, the UAC does not respond to the OPTIONS message even though it was sent multiple times. Because the OPTIONS fails, when the Oracle Enterprise Session Border Controller receives another REGISTER, it responds with the previously successful timer value (in this case, the minimum NAT interval).

However, if the OPTIONS request succeeds, then the Oracle Enterprise Session Border Controller persists with the test until it fails or until the maximum NAT timer value is reached. In this case, when the OPTIONS message fails, the Oracle Enterprise Session Border Controller uses the last successful test-timer value as the time for the expires header in the 200 OK for the REGISTER message.

Synchronize A-HNT Successful Timer to Standby

Adaptive HNT enables the Oracle Enterprise Session Border Controller to determine, through testing, an optimum SIP REGISTER expires time interval that keeps the NAT pinhole open. For an HA node, this successful time value is determined through testing by the active system and then replicated to the standby. If there is a switchover during the active system's testing process, then it will restart for that endpoint.

Adaptive NHT Configuration

You configure the SIP interface to set the state of this feature and to define the increments of time the Oracle Enterprise Session Border Controller uses to perform adaptive HNT. Remember that the Oracle Enterprise Session Border Controller uses the time you specify as the NAT interval, the supported time interval, as the basis on which to begin testing.

To configure adaptive HNT:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ```

4. sip-dynamic-hnt—Enable this parameter if you want to use adaptive HNT. The default value is disabled. The valid values are:

   - enabled | disabled

5. max-nat-interval—Set the amount of time in seconds that testing should not exceed. The Oracle Enterprise Session Border Controller will keep the expires interval at this value. The default value is 3600. The valid range is:

   - Minimum—0
   - Maximum—999999999

6. nat-int-increment—Set the amount of time in seconds to use as the increment in value in the SIP expires header. The default value is 10. The valid range is:
   - Minimum—0
   - Maximum—999999999
7. nat-test-increment—Set the amount of time in seconds that will be added to the test timer. The default value is 30. The valid range is:
   - Minimum—0
   - Maximum—999999999

SIP IP Address Hiding and NATing in XML

Adding to its topology hiding and NAT capabilities, the Oracle Enterprise Session Border Controller now performs those functions for pertinent IP addresses that are not part of the standard SIP message header format. Previously, such addresses were visible to the next hop in the SIP session path.

Note that this feature adds to the Oracle Enterprise Session Border Controller's pre-existing ability to perform this function for XML messages; this new support is specifically for the keyset-info message type.

For incoming SIP NOTIFY messages, the Oracle Enterprise Session Border Controller searches for the application/keyset-info+xml content type in the message. When it finds this content type, it searches further to detect the presence of <di:remote-uri> or <di:local-uri> XML tags and then NATs the IP addresses in the tags it finds. Specifically, the Oracle Enterprise Session Border Controller changes:

- The <di:remote-uri> IP address to be the egress SIP interface's IP address
- The <di:local-uri> IP address to be the Ip address of the next hop to which the message is being sent

Sample SIP NOTIFY with NATed XML

The following is a sample SIP NOTIFY message as it might arrive at the Oracle Enterprise Session Border Controller.

Note: that it contains the <di:remote-uri> or <di:local-uri> XML tags on which the system will perform NAT; these lines appear in bold text.

```
NOTIFY sip:15615281021@10.152.128.253:5137;transport=udp SIP/2.0
To: 15615281021  <sip:15615281021@10.152.128.102:5080>;tag=5c93d019904036a
From: <sip:15615281021@10.152.128.102:5080>;tag=test_tag_0008347766
Call-ID: 3215a76a979d0c6
CSeq: 18 NOTIFY
Contact: <sip:15615281021@10.152.128.102:5080;maddr=10.152.128.102>
Via: SIP/2.0/UDP 10.152.128.102:5060;branch=z9hG4bK_brancha_0023415201
Event: keyset-info
Subscription-state: active;expires=2778
Accept: application/keyset-info+xml
Content-Type: application/keyset-info+xml
Content-Length: 599
Max-Forwards: 70
<?xml version="1.0"?>
<keyset-info xmlns="urn:ietf:params:xml:ns:keyset-info"
     version="16"
     entity="15615281021">
  <ki-data>
    <ki-state>"active"</ki-state>
    <ki-event>"unknown"</ki-event>
  </ki-data>
   <di:dialog id="dialog_id_201" call-
id="1395216611-1987932283256611-11-0884970552" local-
tag="test_tag_0008347790" direction="recipient">
    <di:state>trying</di:state>
    <di:duration>2778</di:duration>
```

```
    <di:local-uri>sip:15615281021@10.152.128.253:5137</di:local-uri>
    <di:remote-uri>sip:1004@10.152.128.102</di:remote-uri>
   </di:dialog>
</keyset-info>
```

Once the Oracle Enterprise Session Border Controller has completed the NAT process, the <di:remote-uri> and <di:local-uri> XML tags look like this

```
<di:local-uri>sip:15615281021@192.168.200.99:5137</di:local-uri>
<di:remote-uri>sip:1004@192.168.200.49</di:remote-uri>
```

because egress the SIP interface's IP address is 192.168.200.49 and the next hop's IP address is 192.168.200.99.

This feature does not require any configuration.

SIP Server Redundancy

This section explains how to configure SIP server redundancy. SIP server redundancy involves detecting that an upstream/downstream SIP signaling entity has failed, and adapting route policies dynamically to remove it as a potential destination.

Overview

You establish SIP server redundancy by creating session agents, which are virtual representations of the SIP signaling entities. These agents are then collected into a session agent group, which is a logical collection of two or more session agents that behaves as a single aggregate entity.

Rather than direct signaling messages to a single session agent (IP), the signaling message is directed to a session agent group (SAG). The group will have a set distribution pattern: hunt, round robin, proportionally distributed, and so on. Signaling is spread amongst the agents using this chosen pattern.

You direct the signaling message by configuring a route policy, known as a local policy, which determines where SIP REQUESTS should be routed and/or forwarded. The values in the To and From headers in the SIP REQUEST are matched with the content of the local policy within the constraints set by the session agent's previous hop value and SIP interface values such as the list of carriers.

To summarize, you need:

- two or more session agents
- a session group containg those session agents
- a local policy which directs traffic to the session agent group

Configuration Overview

You make a session agent group a target by using a local policy to select the next hop from the members of a session agent group. You need to set the replace URI field of the configured local policy to enabled; which causes NAT rules such as realm prefixing to be overridden. The replace URI field allows you to indicate whether the local policy's value is used to replace the Request-URI in outgoing requests. This boolean field can be set to either enabled or disabled.

When the SIP NAT's route home proxy field is set to forced, it forces the Request to be forwarded to the home proxy without using a local policy. When this option is set to either disabled or enabled and the Request-URI matches the external address of the SIP NAT, the local policy is used.

However, the local policy only replaces the Request-URI when the original Request-URI matches the SBC's IP address or hostname. This behavior is in accordance with that described in RFC 3261. The original Request-URI will be the home proxy address value (the home address of the SIP NAT into the backbone) and not the Oracle Enterprise Session Border Controller (E-SBC) address.

Using strict routing, the Request-URI would be the next hop, but the message would also include a Route header with the original Request-URI. With loose routing, the Request-URI remains unchanged and the next hop value is added as the top Route header.

Sometimes the next hop field value must replace the Request-URI in the outgoing request, even if the original Request-URI is not the E-SBCC. To accomplish this, an option has been added to the local policy that causes the next hop value to be used as the Request-URI and prevents the addition of Route headers. This option is the replace uri value in the local policy.

The following table lists the policy attributes for the local policy:

Parameter	Description
next hop	IP address of your internal SIP proxy. This value corresponds to the IP address of the network interface associated with the SIP proxy.
realm	Number of the port associated with the SIP port.
replace uri	Stores the transport protocol used for sending an receiving signaling messages associated with the SIP port.
allow anonymous	Indicates whether this SIP port allows anonymous connections from session agents.

Note: You should also define the ping method intervals for the session agents so that the E-SBC can detect when the agents are back in service after failure.

SIP Server Redundancy Configuration

To enable replace URI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type local-policy and press Enter to access the system-level configuration elements. The system prompt changes.

   ```
   ACMEPACKET(configure)# local-policy
   ACMEPACKET(local-policy)#
   ```

3. Type policy-attributes and press Enter. The system prompt changes.

   ```
   ACMEPACKET(local-policy)# policy-attributes
   ACMEPACKET(local-policy-attributes)#
   ```

   From this point, you can configure policy attributes for the local policy. To see all local policy attribute options, enter a ? at the system prompt.

4. action—Set this parameter to replace-uri, which causes NAT rules such as realm prefixing to be overridden. The default value is none. Valid values are:

   - none | replace-uri | redirect

     The replace URI field allows you to indicate whether the local policy's value is used to replace the Request-URI in outgoing requests. This boolean field can be set to either enabled or disabled.

Administratively Disabling a SIP Registrar

The Oracle Enterprise Session Border Controller's registration cache feature is commonly used to support authorization. It also allows the Oracle Enterprise Session Border Controller to respond directly to SIP REGISTER requests from endpoints rather than forwarding every REGISTER message to the Registrar(s). In the Oracle Enterprise Session Border Controller, Registrars are frequently configured as session agents, and an association between each endpoint and its Registrar is stored with the registration cache information.

In Release 4.0.1 and later, the invalidate-registrations parameter in the session agent configuration enables the Oracle Enterprise Session Border Controller to detect failed Registrar session agents and automatically forward subsequent REGISTER requests from endpoints to a new Registrar. You can now perform the same behavior manually through a

new ACLI command. When you use this command, the Oracle Enterprise Session Border Controller acts as though the registrations have expired.

For each SIP session agent, you can enable the manual trigger command, and then use the command from the main Superuser ACLI prompt. The reset session-agent command provides a way for you to send a session agent offline. Session agents can come back online once they send 200 OK messages the Oracle Enterprise Session Border Controller receives successfully.

Without using the manual trigger, session agents can go offline because of they do not respond to pings or because of excessive transaction timeouts. However, you might not want to use these more dynamic methods of taking session agents out of service (and subsequently invalidating any associated registrations). You can disable both of these mechanisms by setting the following parameters to 0:

- ping-interval—Frequency (amount of time in seconds) with which the Oracle Enterprise Session Border Controller pings the entity the session agent represents)
- ttr-no-response—Dictates when the SA (Session Agent) should be put back in service after the SA is taken OOS (Out Of Service) because it did not respond to the Oracle Enterprise Session Border Controller

However, you can still use the new SIP manual trigger even with these dynamic methods enabled; the trigger simply overrides the configuration to send the session agent offline.

Considerations for Implicit Service Route Use

When implicit service route support is enabled for a SIP interface (in IMS applications), the Oracle Enterprise Session Border Controller stores the Service Route URIs from the Service-Route headers that are included in 200 OK responses to REGISTER messages. Subsequently, and even when a session agent is rendered invalid, re-REGISTER messages follow the route stored in the cache instead of using the one defined in the Oracle Enterprise Session Border Controller.

However, you might not want to use this behavior when you send session agents offline. If you instead want use the route defined in the Oracle Enterprise Session Border Controller, then you need to configure the SIP interface option called route-register-no-service-route.

Manual Trigger Configuration

This section shows you how to enable the manual trigger for sending session agents out of service, and how to then use the trigger from the command line. This section also shows you how to verify that you have successfully put a session agent out of service.

To enable a SIP session agent to manually trigger it to go out of service:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration you want to edit.

4. invalidate-registrations—Set this parameter to enabled if you want to use the manual trigger to send this session agent offline (and therefore invalidate the registrations associated with it). The default is disabled.
5. Save and activate your configuration.

   To use the manual trigger that sends session agents offline:

6. Note the hostname value (typically the IP address of the endpoint) for the session agent you want to put out of service. You use this name as an argument in the ACLI command to use the manual trigger.
7. At the Superuser prompt, type reset session-agent, a Space, and the hostname value for the session agent. The press Enter.

```
ACMEPACKET# reset session-agent 192.168.20.45
```

If you enter a session agent that does not exist, the system notifies you that it cannot carry out the reset.

Manual Trigger Confirmation

To confirm that a session agent has been sent offline:

Use the show sipd endpoint-ip command to confirm the session agent state.

```
ACMEPACKET# show sipd endpoint-ip 1016
User <sip:1016@172.18.1.80>
        Contact exp=3582
            UA-Contact: <sip:1016@192.168.1.132:5060> UDP
                realm=access local=172.18.1.132:5060
UA=192.168.1.132:5060
    SD-Contact: <sip:1016-o3badgbbnjcq5@172.18.2.80:5060> realm=core
    Call-ID: 1-7944@192.168.1.132’
    SR=172.18.2.92
    SA=172.18.2.93
    Service-Route='<sip:test@s-cscf::5060;orig;lr>'
ACMEPACKET# reset session-agent 172.18.2.92
Accepted
Reset SA failover timer
ACMEPACKET# show sipd endpoint-ip 1016
User <sip:1016@172.18.1.80>
        Contact <invalidated> exp=3572
            UA-Contact: <sip:1016@192.168.1.132:5060> UDP
                realm=access local=172.18.1.80:5060
UA=192.168.1.132:5060
    SD-Contact: <sip:1016-o3badgbbnjcq5@172.18.2.80:5060> realm=core
    Call-ID: 1-7944@192.168.1.132’
    SR=172.18.2.92 (failed 2 seconds ago)
    SA=172.18.2.93
    Service-Route='<sip:test@s-cscf::5060;orig;lr>'
ACMEPACKET#
```

In the above ACLI example the first iteration of the show sip endpoint-ip command provides information for the in-service 172.18.2.92 session agent; the second command iteration displays information for the now out-of-service session agent.

Surrogate Agent Refresh on Invalidate

Surrogate agent registrations normally only re-register when nearing their expiration time. When a registrar fails, the surrogate agent will wait until the expiration time to refresh the registration with an in-service registrar.

You can configure your Oracle Enterprise Session Border Controller to immediately refresh the surrogate agent registrar with an in-service registrar by enabling the existing parameter invalidate-registrations.

Invalidate Registrations

An existing feature called invalidate-registrations located in the session agent keeps track of when surrogate agents go out of service. When REGISTER messages are received, registration entries that had out-of-service session agents since the last REGISTER will always allow the message through to the registrar (as opposed to responding directly from the cache).

The invalidate-registrations parameter in session agent configuration enables the Oracle Enterprise Session Border Controller to detect failed Registrar session agents.

If invalidate-registrations is enabled for the session agent, a response from a surrogate REGISTER that contains a service-route header that corresponds to a session-agent is installed to the registration cache entry.

The surrogate-agents are scanned. Surrogate agents with registration entries matching the out-of-service registrar have their timer reset to initiate a refresh. For an immediate refresh, the registration entry will only be considered when the service-route session agent goes out-of-service. The service-route session agent takes precedence and any previous registrar session agent w ill not be considered for an immediate refresh of the surrogate-agent registration.

Performance Impact

In cases with a large number of surrogate-agent registrations, there may be an impact to CPU usage when a session-agent goes out-of-service. All of the surrogate-agent registrations are scanned at that time. Refresh registrations are then sent out on timers.

Media Inactivity Timer Configuration

To disable the media inactivity timer for calls placed on hold:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration you want to edit.

4. invalidate-registration—Set this parameter to enabled if you want to use the manual trigger to send this session agent offline (and therefore invalidate the registrations associated with it). The default is disabled.
5. Save and activate your configuration.

SIP Distributed Media Release

This section explains how to configure distributed media release (DMR). SIP DMR lets you choose whether to include multi-system (multiple Oracle Enterprise Session Border Controllers) media release information in SIP signaling requests sent into a specific realm.

Overview

The SIP DMR feature lets RTP/RTCP media be sent directly between SIP endpoints (for example, SIP phones or user agents) without going through a Oracle Enterprise Session Border Controller; even if the SIP signaling messages traverse multiple Oracle Enterprise Session Border Controllers. It encodes IPv4 address and port information for the media streams described by the media, for example SDP.

With SIP DMR, the media realm and IPv4 address and port information from the UA's SDP is encoded into SIP messages (either in the SIP header or in the SDP) as they enter the backbone network. The information is decoded by a Oracle Enterprise Session Border Controller from SIP messages that come from the backbone network. The decoded address and port information is put into the SDP sent the UAs in the access (private/customer) network.

This functionality lets the RTP/RTCP flow directly between the UAs in the access network without traversing the Oracle Enterprise Session Border Controllers and without passing into the backbone network. The media can then

flow directly between the two SIP endpoints in the same network, if it is serviced by multiple Oracle Enterprise Session Border Controllers.

You can enable this feature on a per-realm basis and multiple realms can be supported.

Endpoint Locations

You can configure the Oracle Enterprise Session Border Controller to release media when the source and destination of the call are in the same network, customer VPN, or customer LAN. In architectures that use DMR, the Oracle Enterprise Session Border Controller is only part of the media path for traffic that originates and terminates in different networks.

If configured to do so, the Oracle Enterprise Session Border Controller can release media:

- Between endpoints supported by a single Oracle Enterprise Session Border Controller

  In the same network/VPN

  In the same network behind the same NAT/firewall
- Between endpoints supported by multiple distributed Oracle Enterprise Session Border Controllers

  In the same network/VPN

Location of the Encoded Information

Encoded media release information can appear in three different places:

- SDP attribute

  Media release data can be encoded into an SDP attribute in the SIP message body (for example, media-release=sdp;acme-media). The encoded data is placed into an acme-media attribute in the SDP:

  ```
  a=acme-media:<encoded-media-interface-info>
  ```
- SIP header parameter

  Media release data can be placed in a header parameter of a SIP header (for example, media-release=Contact;acme-media). The encoded data is placed into an acme-media parameter in the Contact header:

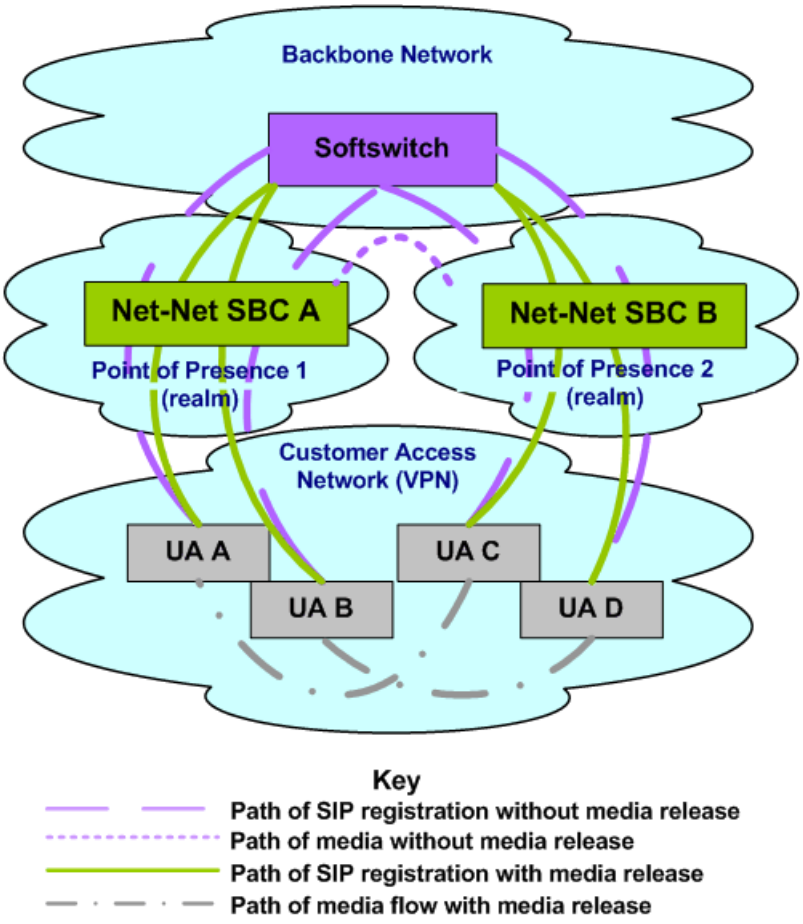
  ```
  Contact: <sip:1234@abc.com>;acme-media=<encoded-media-interface-info>
  ```
- SIP header

  Media release data can appear in a SIP header (for example, media-release=P-Media-Release). The encoded data is placed into a P-Media-Release header:

  ```
  P-Media-Release: <encoded-media-interface-info>
  ```

Example Distributed Media Release

The following example shows the network diagram for DMR in a multiple-site VPN environment supported by multiple, distributed Oracle Enterprise Session Border Controllers.



As shown in the network diagram, UA A and UA B register with the softswitch through Oracle Enterprise Session Border Controller A while UA C and UA D register with the softswitch through Oracle Enterprise Session Border Controller B. Without DMR, the media for calls between UA A/UA B and UA C/UA D is steered through both Oracle Enterprise Session Border Controller A and Oracle Enterprise Session Border Controller B.

With SIP DMR, the media realm and IPv4 address and port information from the UA's Session Description Protocol (SDP) is encoded into SIP messages (either in the SIP header or in the SDP) as they enter the backbone (public/service provider) network. The information is decoded from SIP messages that come from the backbone network. The decoded address and port information is put into the SDP sent to the UAs in the access (private/customer) network. This functionality allows for the RTP/RTCP to flow directly between the UAs in the access network without traversing the Oracle Enterprise Session Border Controllers and without passing into the backbone network.

Overview of SIP DMR Configuration

To configure SIP DMR:

1. Edit the SIP config element's option field.

   The media-release="<header-name>[;<header-param>]" option defines how the SIP distributed media release feature encodes IPv4 address and port information. If the media-release parameter is configured in the options field but no header is specified, the parameter value of P=Media-Release will be used. This parameter is optional and is not configured by default.

2. Enable SIP DMR for the entire realm by setting the realm config element's msm release field to enabled.

   The media IPv4 address and port information is encoded into outgoing SIP messages and decoded from incoming SIP messages for all of the realms (in each realm-config element) with which the SIP distributed media release will be used.

Note: You can also use the realm config element's mm in network field to release the media back to a connected network that has multiple realms. This field is not specific SIP distributed media release and it is not required for the SIP DMR to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Oracle Enterprise Session Border Controller release the media.

SIP DMR Configuration

To configure media release:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

   From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.

4. Type options followed by a Space.
5. After the Space, type the media release information in the following format:

```
media-release="<header-name>[;<header-param>]"
```

   - header-name either refers to the SIP header in which to put the information or to the special header-name value of sdp to indicate the information should be put into the SDP.
   - parameter-name refers to the header parameter name in which to put the information or, in the case of the special header-name value of sdp, to the SDP attribute name in which to put the information.

     For example:

     ```
     ACMEPACKET(sip-config)# options media-release=P-Media-Release
     ```

6. Press Enter.

   Note: If the media-release parameter is configured in the options field, but no header is specified, then the parameter value of P-Media-Release will be used. P-Media-Release is a proprietary header and means that the media will be encoded in the SIP header with this name.

   The following example shows where the encoded information (for example, SDP data) is passed.

```
media-release="P-Media-Release"
media-release="Contact;acme-media"
media-release="sdp;acme-media"
```

Configuring the Realm

You need to set the each realm config element's msm release field to enabled for all the realms for which you want to use SIP DMR.

Although the mm in network field is not specific to the SIP distributed media release feature, it can be used to release the media back to a connected network that has multiple realms. This field does not need to be configured in order for the SIP distributed media release feature to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Oracle Enterprise Session Border Controller release the media.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

4. msm-release—Enable DMR within this realm on this Oracle Enterprise Session Border Controller. The default value is disabled. The valid values are:

   - enabled | disabled

5. Repeat for each realm on which you want to enable DMR.

Add-On Conferencing

This section explains how to configure the add-on conferencing functionality. It also includes a description of the SIP B2BUA functionality related to the SIP add-on conferencing. This description includes information about Contact header mapping and processing and Refer-to header processing.

Overview

SIP add-on conferencing lets you:

- Use the Oracle Enterprise Session Border Controller's add-on conferencing feature for network architectures in which the conference initiator is located on a different network than that of the media server.
- Configure the Oracle Enterprise Session Border Controller to enable Contact header mapping for the Refer-To header.

Caveats

The following caveats are associated with add-on conferencing:

- Contact header mapping is not replicated on the standby Oracle Enterprise Session Border Controller in an HA Oracle Enterprise Session Border Controller pair architecture.
- Upon switchover, any conferences in progress remain in progress, but no new parties can be invited to or join the conference.
- By default, the Oracle Enterprise Session Border Controller does not map SIP Contact headers for reasons of performance.

Add-On Conferencing Scenario

The add-on conferencing scenario described in the following example applies to a network architecture involving the Oracle Enterprise Session Border Controller and a media server that is located on a different network from the other conference participants. In this scenario, the Oracle Enterprise Session Border Controller resides on a standalone network that connects two additional, separate networks.

Some network architectures have a media server on a different network from the one on which the phones reside. In this scenario, all requests and/or responses going from the phones (Phone A, Phone B, or Phone C) to Media Server D and vice versa are translated according to their corresponding SIP-NAT. All headers subjected to NAT are encoded and decoded properly as they traverse the Oracle Enterprise Session Border Controller, except for the Contact header. This exception occurs because the SIP process on the Oracle Enterprise Session Border Controller runs as a SIP B2BUA and not as a SIP proxy.

The SIP B2BUA re-originates the Contact headers of the User Agents (UAs) participating in SIP sessions with local Contact headers to make sure that they receive all future in-dialog requests. For an in-dialog request, the B2BUA can identify the dialog and find the Contact URI of the other leg of the call.

The Oracle Enterprise Session Border Controller add-on conferencing feature applies to situations when the Contact URI is used in another dialog. In such a case, the SIP B2BUA will not be able to find the correct dialog that retrieves the correct Contact URI of the other leg if it needs to replace the Contact URI.

Using the SIP add-on conferencing, the SIP B2BUA on the Oracle Enterprise Session Border Controller can map the Contact headers it receives to the Contact headers it creates. It can also convert the Refer-To URI to the correct value required for forwarding the REFER request.

SIP B2BUA Functionality

This section describes the role of the Oracle Enterprise Session Border Controller's SIP B2BUA in the add-on conferencing scenario that requires Contact header mapping for the Refer-To header.

When the Oracle Enterprise Session Border Controller starts up, the SIP B2BUA reads and parses the list of options in the SIP configuration. If the refer to uri prefix is an appropriate value (it is not an empty string), the Oracle Enterprise Session Border Controller will have a text prefix value the media server can use to denote a conference ID in its Contact header. With this information, the SIP B2BUA sets up a Contact header mapping.

You configure the Oracle Enterprise Session Border Controller to enable Contact header mapping for the Refer-To header by editing the SIP config options parameter. The SIP B2BUA on the Oracle Enterprise Session Border Controller can then map the Contact headers it receives to the Contact headers it creates.

Contact Header Processing

The Contact header mapping matches a Contact header that contains the refer to URI prefix to the corresponding Contact header that the Oracle Enterprise Session Border Controller's SIP B2BUA re-originates. Contact headers that do not contain the refer to URI prefix are not mapped (so that performance of the Oracle Enterprise Session Border Controller is minimally affected).

Only the Contact header in an INVITE request and its 200 OK response are checked for the refer to URI prefix and added to the Contact header mapping. Contact headers appearing in other SIP requests/responses are not checked.

Target Mapping and Conferences

If the Oracle Enterprise Session Border Controller is configured to enable Contact header mapping for the Refer-To header, then Contact header target maps are established for each individual call. The Oracle Enterprise Session Border Controller's SIP B2BUA uses these maps to allow the media server to connect the conference initiator with the conferenced-in parties.

Prior to terminating the call (hanging up), the conference initiator can contact other parties and invite those additional parties to join the conference. These other parties can join the existing conference because the target mapping for the conference is still in effect on the Oracle Enterprise Session Border Controller.

Once the conference initiator hangs up, the Oracle Enterprise Session Border Controller discards the mapping from the conference.

Refer-To Header Processing

When a Refer-To header is present in a REFER request that arrives at the SIP B2BUA after the incoming request is properly translated according to its SIP-NAT, the SIP B2BUA follows these steps:

1. The SIP B2BUA parses the Refer-To URI.
2. If the user part of the Refer-To URI contains the refer to URI prefix, the SIP B2BUA searches the Contact header mapping for a match of the user part of the URI.

   If the user part of the Refer-To URI does not contain the refer to URI prefix, the SIP B2BUA leaves the existing Refer-To URI unchanged.
3. If the user part of the Refer-To URI contains the refer to URI prefix and a match of the Refer-To URI is found, the SIP B2BUA replaces the existing Refer-To URI with the URI of the corresponding Contact URI stored in the matched record. This replacement enables the NAT function to properly decode the replacement URI and change it back to the form originally received by the Oracle Enterprise Session Border Controller. As a result, the correct conference ID is restored in the Refer-To header prior to the request being sent to its next hop.

If the user part of the Refer-To URI contains the refer to URI prefix but a matched URI cannot be found, the SIP B2BUA will leave the existing Refer-To URI unchanged and will write a WARNING level log message to record the failure.

Add-on Conferencing Configuration

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.

4. Type options followed by a Space.
5. After the Space, type the add-on conferencing information in the following format:

   ```
   options refer-to-uri-prefix=”conf=”
   ```

   For example:

   ```
   ACMEPACKET(sip-config)# options refer-to-uri-prefix=”conf”
   ```

6. Press Enter.

SIP REFER Method Call Transfer

In prior releases, the Oracle Enterprise Session Border Controller supports the SIP REFER method by proxying it to the other UA in the dialog. A handling mode has been developed for the REFER method so that the Oracle Enterprise Session Border Controller automatically converts a received REFER method into an INVITE method, thus allowing the Oracle Enterprise Session Border Controller to transfer a call without having to proxy the REFER back to the other UA.

This function can be configured for a specified SIP interface, a realm, or a session agent. When all three elements have the SIP REFER method call transfer functionality configured, the session-agent configuration takes precedence over realm-config and sip-interface configurations. If session-agent is not configured, and realm-config and sip-interface are, realm-config takes precedence.

The Oracle Enterprise Session Border Controller has a configuration parameter giving it the ability to provision the handling of REFER methods as call transfers. The parameter is called refer-call-transfer. When this feature is enabled, the Oracle Enterprise Session Border Controller creates an INVITE message whenever it receives a REFER. The Oracle Enterprise Session Border Controller sends this INVITE message to the address in the Refer-To header. Included in the INVITE message is all the unmodified information contained in the REFER message. The previously negotiated codec is also still used in the new INVITE message. NOTIFY and BYE messages are sent to the UA upon call transfer completion.

If a REFER method is received containing no Referred-By header, the Oracle Enterprise Session Border Controller adds one, allowing the Oracle Enterprise Session Border Controller to support all call agent screen applications.

In addition, the SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers
- Both successful and unsuccessful call transfers
- Early media from the Referred-To party to the transferee
- REFER method transfer from different sources within the destination realm

- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.
- The Referred-To party can be both in a different realm (and thus a different steering pool) from the referrer, and in the same realm
- The associated latching should not prohibit the Referred-To party from being latched to while the referee is still sending media.

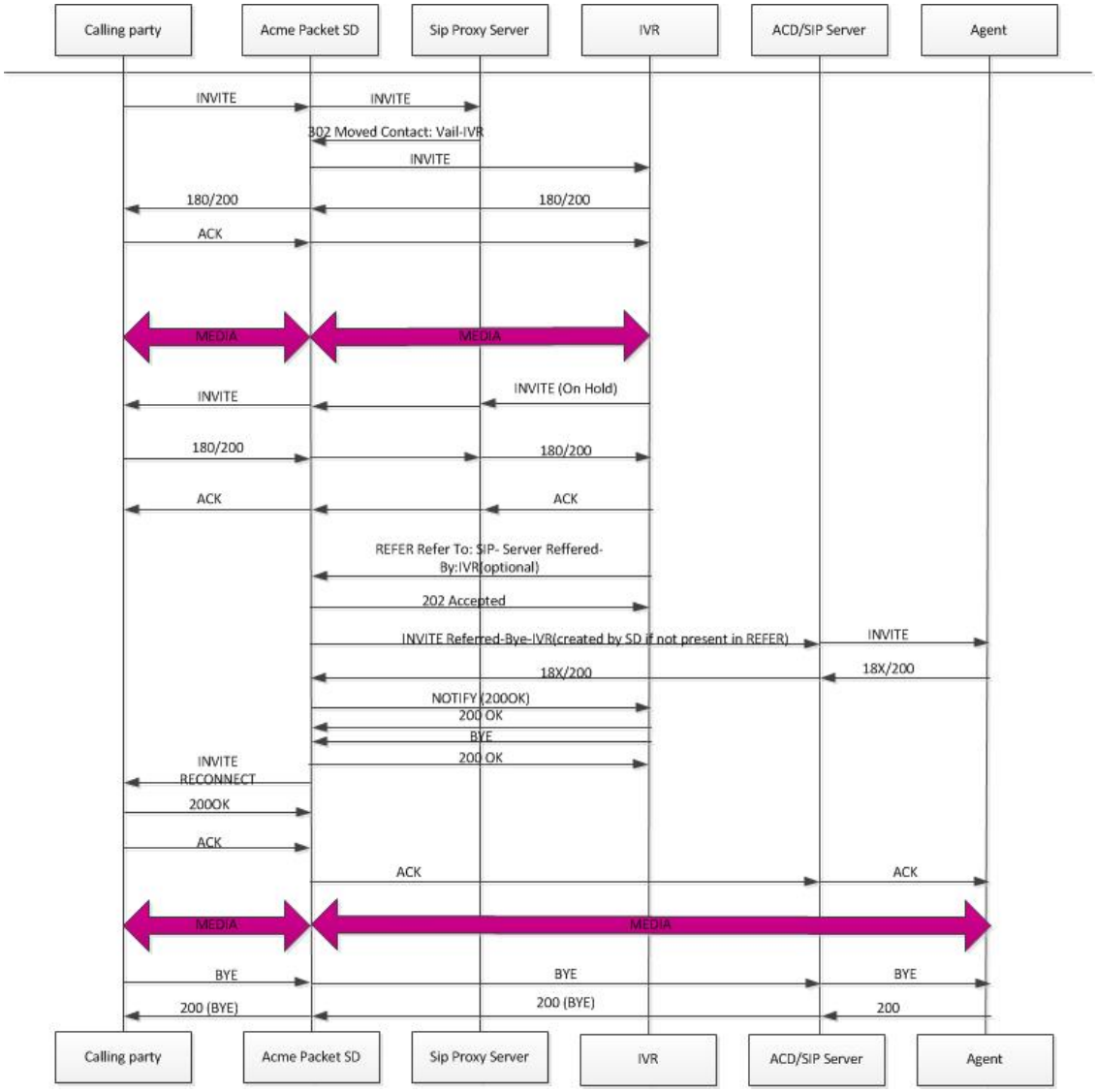
Unsuccessful Transfer Scenarios

The Oracle Enterprise Session Border Controller does not successfully handle the following failed, unusual, and unexpected transfer scenarios:

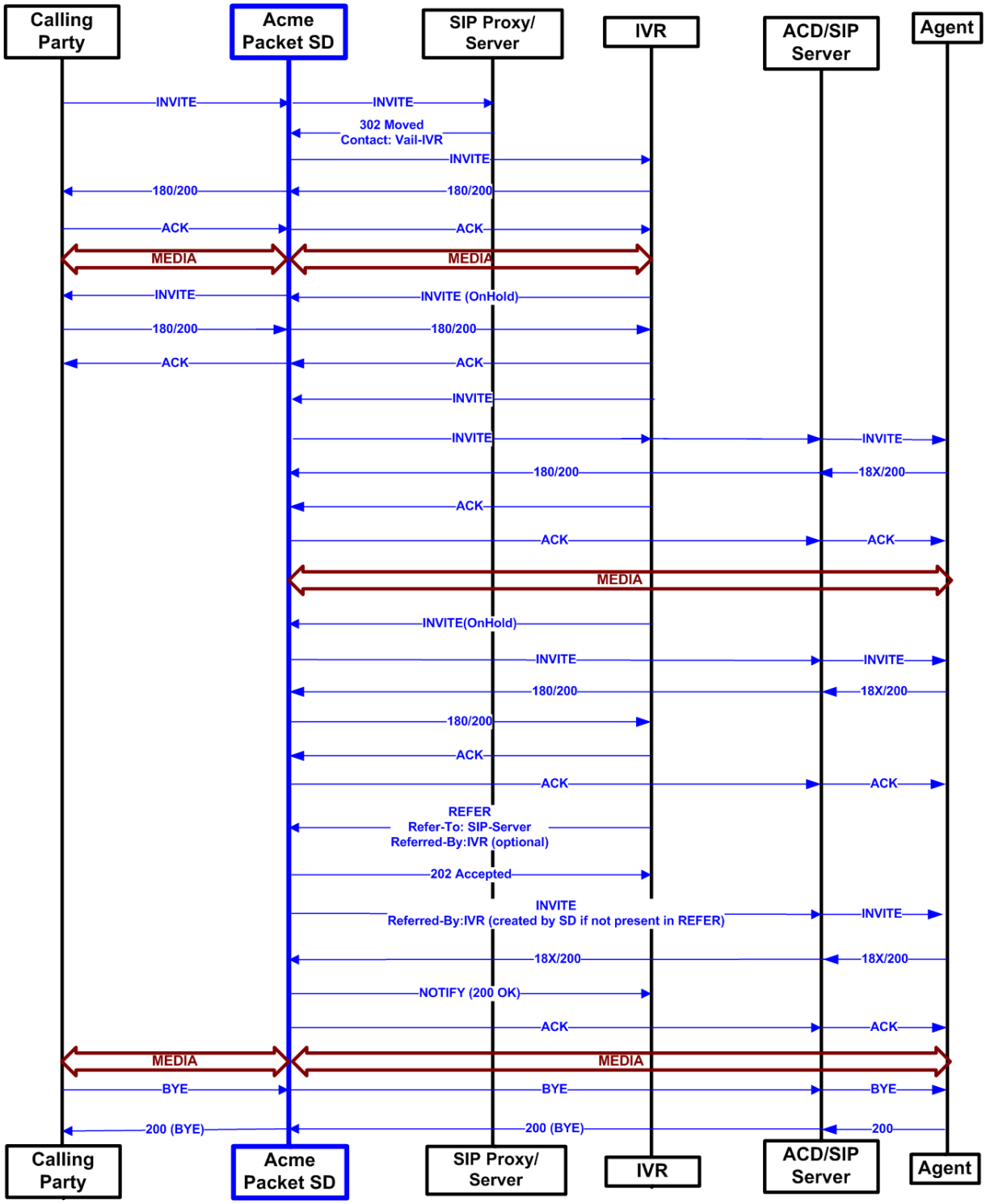
- The new INVITE to the Referred-To party gets challenged, the Oracle Enterprise Session Border Controller does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Oracle Enterprise Session Border Controller.
- The Oracle Enterprise Session Border Controller shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Oracle Enterprise Session Border Controller ignores any MIME attachment(s) within a REFER method.
- The Oracle Enterprise Session Border Controller recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.
- The original parties agreed on a codec using a dynamic payload type, and the Referred-To party happens to use a different dynamic payload number for that codec.

Call Flows

The following is an example call flow for an unattended call transfer:



The following is an example call flow of an attended call transfer:



SIP REFER Method Configuration

To enable SIP REFER method call transfer in the realm-config:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. refer-call-transfer—Set to enabled to enable the refer call transfer feature. The default for this parameter is disabled.
5. Save and activate your configuration.

To enable SIP REFER method call transfer in the sip-interface:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-config)#
```

9. refer-call-transfer—Set to enabled to enable the refer call transfer feature. The default for this parameter is disabled.
10. Save and activate your configuration.

To enable SIP REFER method call transfer in a realm:

11. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

12. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

13. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

14. refer-call-transfer—Set to enabled to enable the refer call transfer feature. The default for this parameter is disabled.
15. Save and activate your configuration.

To enable SIP REFER method call transfer in the session-agent:

16. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

17. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

18. Type session-agent and press Enter.

```
ACMEPACKET(media-manager)# session-agent
ACMEPACKET(session-agent)#
```

19. refer-call-transfer—Set to enabled to enable the refer call transfer feature. The default for this parameter is disabled.
20. Save and activate your configuration.

REFER-Initiated Call Transfer

In prior releases, the Oracle Enterprise Session Border Controller supported REFER-initiated call transfer either by proxying the REFER to the other User Agent in the dialog, or by terminating the received REFER and issuing a new INVITE to the referred party. These static alternate operational modes could be configured for specific SIP interfaces, realms, or session agents.

Release S-C6.2.0 enhances support with an additional operational mode that determines on a call-by-call basis whether to proxy the REFER to the next hop, or terminate the REFER and issue an INVITE in its stead.

Note: With the release of Version S-C6.2.0, support for REFER-initiated call transfer is no longer available for SIP interfaces; support must be configured for realms and/or session agents.

Version S-C6.2.0 provides a new configuration parameter dyn-refer-term, and a revised refer-call-transfer parameter (both available in realm-config configuration mode) that specify call transfer modes.

With the refer-call-transfer parameter set to disabled (the default), all received REFERs are simply proxied to the peer User Agent.

With the refer-call-transfer parameter set to enabled, the Oracle Enterprise Session Border Controller terminates all REFERs, generates a new INVITE, and sends the INVITE to the address in the Refer-To header.

With the refer-call-transfer parameter set to dynamic (a new value introduced with Version S-C6.2.0), the Oracle Enterprise Session Border Controller determines REFER handling on a call-by-call basis as follows:

1. Check the refer-call-transfer value for the session agent from which the REFER was received, or for ingress realm (the realm that received the REFER).

   If the value is disabled, proxy the REFER to the peer User Agent, to complete REFER processing.

   If the value is enabled, terminate the REFER and issue an new INVITE to the referred party, to complete REFER processing.

   If the value is dynamic, identify the next hop session agent or the egress realm.
2. Check the dyn-refer-term value for the next hop session agent, or for the egress realm.

   If the dyn-refer-term value is disabled (the default), proxy the REFER to the next hop to complete REFER processing.

   If the dyn-refer-term value is enabled, terminate the REFER and issue an new INVITE to the referred party to complete REFER processing

Supported Scenarios

In the basic scenario for REFER initiated call transfer, a call is established between two User Agents (Alice and Bob). User Agent Bob then sends a REFER request to transfer the call to a third User Agent Eva. With dynamic call-transfer enabled, the Oracle Enterprise Session Border Controller (E-SBC) prevents the REFER from being sent to Alice and generates the INVITE to Eva.

If the INVITE to Eva succeeds, the E-SBC sends a re-INVITE to Alice modifying the SIP session as described in Section 14 of RFC 3261, *SIP: Session Initiation Protocol*. At this point the E-SBC cancels the original dialog between the E-SBC and Bob.

If the INVITE to Eva fails, call disposition depends on whether or not Bob issued a BYE after the REFER call transfer. If the Oracle Enterprise Session Border Controller did receive a BYE from Bob (for instance, a blind transfer), it proxies the BYE to A. Otherwise, the E-SBC retains the original SIP session and media session, thus allowing Bob to re-establish the call with Alice by sending a re-INVITE. In this case, the E-SBC sets a timer (32 seconds), after which a BYE will be sent.

If a REFER method is received containing no Referred-By header, the E-SBC adds one, allowing the E-SBC to support all call agent screen applications.

In addition, the SIP REFER method call transfer feature supports the following:

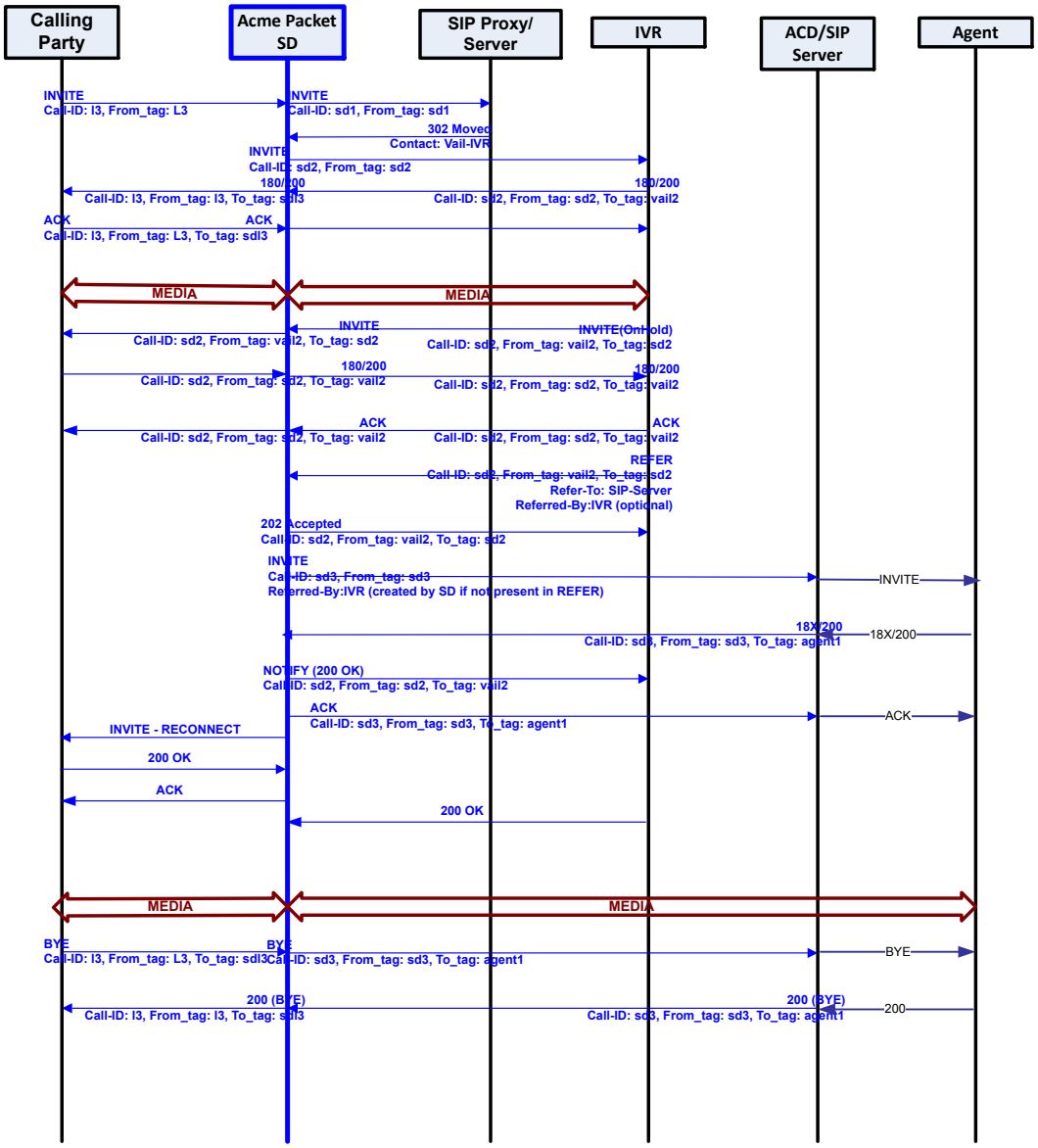
- Both unattended and attended call transfers
- Both successful and unsuccessful call transfers
- Early media from the Referred-To party to the transferee
- REFER method transfer from different sources within the destination realm
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.
- The Referred-To party can be both in a different realm (and thus a different steering pool) from the referrer, and in the same realm
- The associated latching should not prohibit the Referred-To party from being latched to while the referee is still sending media.

The E-SBC does not successfully handle the following anomalous transfer scenarios:

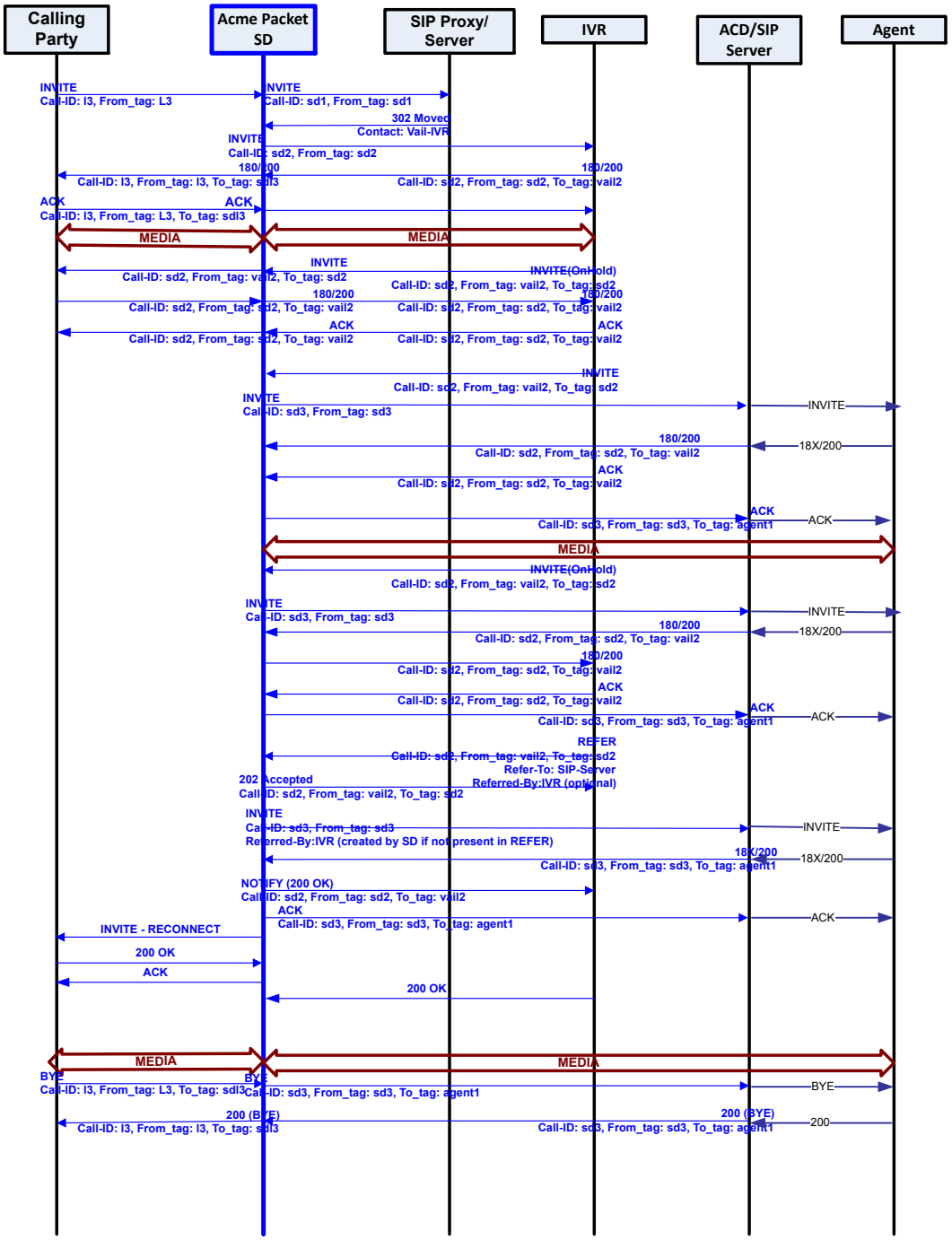
- The new INVITE to the Referred-To party gets challenged — the E-SBC does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the E-SBC.
- The E-SBC shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The E-SBC ignores any MIME attachment(s) within a REFER method.
- The E-SBC recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.

Call Flows

The following is an example call flow for an unattended call transfer:



The following is an example call flow of an attended call transfer:



REFER Source Routing

If, after the conclusion of static or dynamic REFER handling, the REFER is terminated and a new INVITE issued, users now can specify a policy lookup behavior based upon either the source realm of the calling party (the INVITE originator), or the source realm of the referring party (the REFER originator).

Behavior is controlled by a new refer-src-routing parameter in the sip-config configuration element.

disabled, the default value, specifies that the Oracle Enterprise Session Border Controller performs a policy lookup based on the source realm of the calling party.

enabled specifies that the Oracle Enterprise Session Border Controller performs a policy lookup based on the source realm of the referring party.

REFER Source Routing Configuration

To enable realm-based REFER method call transfer:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. refer-call-transfer — Retain the default (disabled) to proxy all REFERs to the next hop. Use enabled to terminate all REFERs and issue a new INVITE. Use dynamic to specify REFER handling on a call-by-call basis, as determined by the value of the dyn-refer-term parameter.
5. dyn-refer-term (meaningful only when refer-call-transfer is set to dynamic) — Retain the default (disabled) to terminate the REFER and issue a new INVITE. Use enabled to proxy the REFER to the next hop.
6. Save and activate your configuration.

To enable session-agent-based REFER method call transfer:

7. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

8. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

9. Type session-agent and press Enter.

```
ACMEPACKET(media-manager)# session-agent
ACMEPACKET(session-agent)#
```

10. refer-call-transfer — Retain the default (disabled) to proxy all REFERs to the next hop. Use enabled to terminate all REFERs and issue a new INVITE. Use dynamic to specify REFER handling on a call-by-call basis, as determined by the value of the dyn-refer-term parameter.
11. Save and activate your configuration.

To specify policy lookup for a newly generated INVITE:

12. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

13. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

14. Type sip-config and press Enter.

```
ACMEPACKET(configure)# sip-config
ACMEPACKET(sip-config)#
```

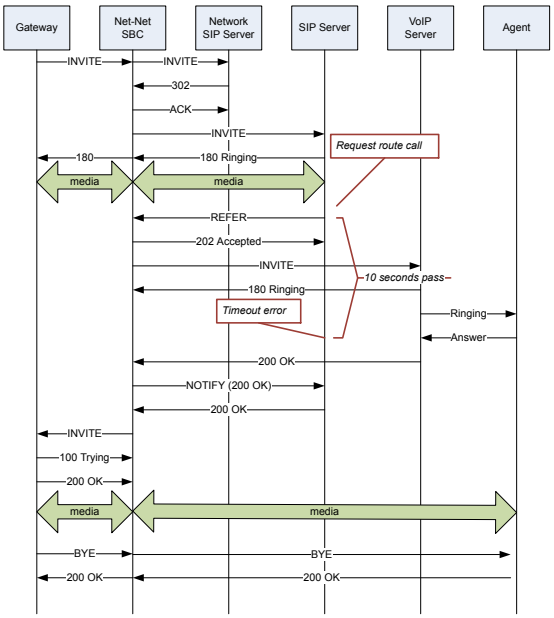
15. refer-src-routing — Retain the default (disabled) to perform a policy lookup based upon the source realm of the calling party (the issuer of the original INVITE). Use enabled to perform a policy lookup based upon the source realm of the referring party (the issuer of the REFER).
16. Save and activate your configuration.

180 & 100 NOTIFY in REFER Call Transfers

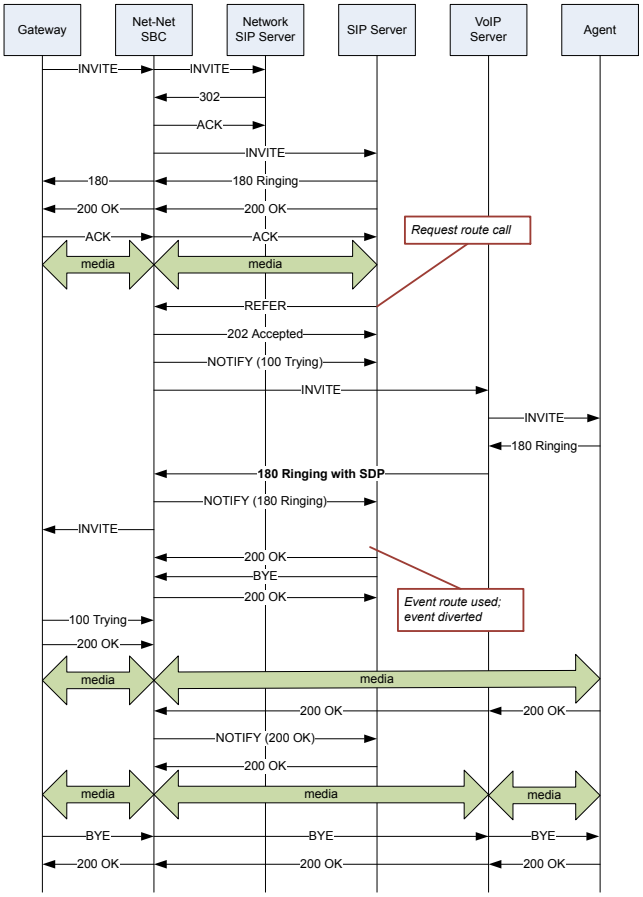
When you configure your Oracle Enterprise Session Border Controller to support REFER call transfers, you can enable it to send a NOTIFY message after it has sent either a 202 Accepted or sent a 180 Ringing message. If your network contains elements that comply with RFC 5589, and so expect the NOTIFY message after the 202 Accepted and each provisional 180 Ringing, you want to set the refer-notify-provisional to either initial or all, according to your needs.

Without this parameter changed from its default (none), the Oracle Enterprise Session Border Controller does not return send the NOTIFY until it receives the 200 OK response from the agent being called. If the time between the REFER and the NOTIFY exceeds time limits, this sequencing can cause the Oracle Enterprise Session Border Controller's NOTIFY to go undetected by devices compliant with RFC 5589. Failures during the routing process can result.

You can see how a sample call flow works without setting the refer-notify-provisional parameter.



When you compare the call flow above to the one depicting the scenario when the Oracle Enterprise Session Border Controller has the refer-notify-provisional changed from its default, you can see that the Oracle Enterprise Session Border Controller now response with a NOTIFY in response to the 202 Accepted and it sends another after the 180 Ringing.This causes the event to be diverted successfully.



Sample Messages

In compliance with RFC 5589, the NOTIFY message with 100 Trying as the message body looks like the sample below. Note that the expires value in the subscription state header is populated with a value that equals 2* TIMER C, where the default value of TIMER C is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
```

```
Content-Length: ...
SIP/2.0 100 Trying
```

Also in compliance with RFC 5589, the NOTIFY message with 180 Ringing as the message body looks like the sample below. Again, the expires value in the subscription state header is populated with a value that equals 2* TIMER C, where the default value of TIMER C is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 180 Ringing
```

Also in compliance with RFC 5589, the NOTIFY message with 200 OK as the message body looks like the sample below.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 74 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: terminated;reason=noresource
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 200 OK
```

180 and 100 NOTIFY Configuration

You can apply the refer-notify-provisional setting to realms or to session agents. This section shows you how to apply the setting for a realm; the same steps and definitions apply to session agents.

If you do not want to insert NOTIFY messages into the exchanges that support REFER call transfers, you can leave the refer-notify-provisional set to none. This means that the Oracle Enterprise Session Border Controller will send only the final result NOTIFY message. Otherwise, you want to choose one of the two settings described in the instructions below.

To enable 100 and 180 NOTIFY messages in REFER call transfers:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. refer-notify-provisional—Choose from one of the following settings, where the Oracle Enterprise Session Border Controller:
   - initial—Sends an immediate 100 Trying NOTIFY, and the final result NOTIFY
   - all—Sends an immediate 100 Trying NOTIFY, plus a notify for each non-100 provisional messages the Oracle Enterprise Session Border Controller receives; and the final result NOTIFY

   ```
   ACMEPACKET(realm-config)# refer-notify-provisional all
   ```
5. Save your work.

SIP REFER Re-Invite for Call Leg SDP Renegotiation

Enhancing the original implementation of SIP REFER termination introduced in Release S-C6.0.0, this change to Oracle Enterprise Session Border Controller behavior allows for SDP renegotiation between both parties of a transferred call.

Scenario

In a call transfer initiated by SIP REFER, a call is established between two user agents, UA-A and UA-B. UA-B then sends a REFER request to transfer the call to UA-C. The challenge is that UA-A and UA-B had already been communicating using mutually agreed-on codec, while UA-C might not be using an entirely different codec.

To solve this problem, the Oracle Enterprise Session Border Controller causes a new SIP session and new media session to be created between UA-A and UA-C. The Oracle Enterprise Session Border Controller removes any resources allocated for use between UA-A and UA-B, and then severs its connection with UA-B. The session between UA-A and UA-C continues.

Alterations to SIP REFER

The original implementation of the SIP REFER feature made available in Oracle Enterprise Session Border Controller Release S-C6.0.0 resulted in instances where SDP parameters were not being communicate properly. Issues arose when the Oracle Enterprise Session Border Controller maintained the original dialog with the user agent that did not support REFER and failed to communicate and SDP changes to that endpoint.

The alterations to SIP REFER made available in Oracle Enterprise Session Border Controller Release S-C(X)6.1.0M2 solve those issues. Now, the Oracle Enterprise Session Border Controller sends re-INVITE with the negotiated SDP to the user agent for which the Oracle Enterprise Session Border Controller performs the call transfer.

Implementation Details

This section describes the details of how the Oracle Enterprise Session Border Controller behaves in SIP REFER scenarios with the changes made in Oracle Enterprise Session Border Controller Release S-C(X)6.1.0M2. The Oracle Enterprise Session Border Controller makes the new call between Party A and Party C appear as though A were participating to allow the Oracle Enterprise Session Border Controller's natural media setup occur in the same way as if the REFER had actually been sent to A and A had sent a new INVITE.

When the Oracle Enterprise Session Border Controller receives a REFER request and determines it needs to handle it locally, it creates a new INVITE made to look like one from Party A. And the Oracle Enterprise Session Border Controller actually processes this INVITE as though it were from Party A. As a result, new SIP and new media sessions are created with new media ports for Parties A and C. When the INVITE to Party C receives a final response, the Oracle Enterprise Session Border Controller sends the result to Party B using a SIP NOTIFY request.

If the new INVITE succeeds, the old context and flows disappear and the new context and flows for the A-to-C connection remain in place. Because of the new media ports, the Oracle Enterprise Session Border Controller sends a re-INVITE to Party A, directing media to the new port and forwarded to Party C. Next, the original dialog with Party B needs to be terminated; if the Oracle Enterprise Session Border Controller has not received Party B's BYE, it will wait five second and then send Party B a BYE.

If the INVITE to Party C fails, the new SIP and media sessions are deleted as are the new context and flows. The Oracle Enterprise Session Border Controller treats Party A differently depending on whether or not a BYE was received from Party B. If a BYE was received from Party B. then the Oracle Enterprise Session Border Controller sends a BYE to Party A. If not, the original SIP and media sessions as well as the context and media flows remain in tact. This way, Party B can re-establish the call with Party A using a re-INVITE. In the case, the Oracle Enterprise Session Border Controller waits 32 second before sending a BYE.

If the Oracle Enterprise Session Border Controller receives a BYE while processing the INVITE to Party C, it sends a CANCEL message to Party C in an attempt to cancel the call. The BYE passes to Party B, and associated sessions, contexts, and flows terminate normally. Still, the Oracle Enterprise Session Border Controller waits for the final response to the INVITE to Party C. If the Oracle Enterprise Session Border Controller receives a successful response, it sends an ACK and then a BYE to terminate the abandoned call. If the Oracle Enterprise Session Border Controller receives an unsuccessful final response, it uses its normal response error handling processes. In either of these last two cases, all sessions, context, and flow are deleted.

Please note that the Oracle Enterprise Session Border Controller does not remove the a=sendonly attribute from the SDP it sends to Party A during the A-to-B call, and extra media ports are not allocated for the original media session.

SIP REFER with Replaces

To support enterprise and call center applications, the Oracle Enterprise Session Border Controller provides the ability for one party participating in a three-way call to request direct connectivity between the other two parties and to leave the call silently when that connectivity is established. SIP supports this function using the Replaces header in a REFER message, also known as REFER with Replaces.

The most common application of REFER with Replaces handling occurs in a high-level sequence like this:

1. The customer calls a customer service line and reaches—via the Oracle Enterprise Session Border Controller—an IVR/ACD (Interactive Voice Response system/Automatic Call Distribution system). In some architectures, these are two separate elements.
2. Based on the customer's selection from the menu of options, the IVR/ACD contacts an agent via the Oracle Enterprise Session Border Controller.
3. Since the ultimate goal is for the IVR/ACD to drop out of the path, it sends a REFER with Replaces to the Oracle Enterprise Session Border Controller. This message indicates the Oracle Enterprise Session Border Controller should replace the IVR/ACD endpoint in the call leg with the agent's endpoint.
4. The Oracle Enterprise Session Border Controller processes the REFER with Replaces, issuing ReINVITEs to the customer with the agent's parameters.
5. The IVR/ACD drops out of the media path once the bridged call between the customer and the agent is established.

   Note that direct media connectivity between endpoints must be possible in order for the REFER with Replaces to be carried out properly. For example, if both endpoints (such as the customer and agent from the example above) are behind the same firewall, direct media connectivity should be possible. However, if one endpoint is behind a firewall and the other is not, then direct media connectivity may not be possible.

   For licensing capacity purposes, note that a bridged session counts as a single call.

SIP REFER with Replaces Configuration

You enable SIP REFER with Replaces handling either in the realm configuration or in the session agent configuration. This section show you how to configure the feature for session agent, though the steps are the same for adding this feature to a realm.

To enable sending ReINVITES to a referred agent on an existing session/dialog:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration that you want to edit.

4. options—Set the options parameter by typing options, a Space, and then the option name refer-reinvite. Then press Enter.

```
ACMEPACKET(session-agent)# options +refer-reinvite
```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

SIP REFER-to-BYE

The Oracle Enterprise Session Border Controller's SIP REFER-to-BYE capability addresses situations when other network elements do not support the REFER method but do offer blind transfer in a SIP BYE request. The target number is encoded in a Reason header of the BYE request. In such cases, the Oracle Enterprise Session Border Controller terminates the REFER and passes the Refer-To number in a Reason header of the BYE.

You configure both SIP interfaces and SIP session agents with the refer-to-bye option to use this function:

- SIP interface—You add this ability to SIP interfaces facing the SIP elements that need to receive a BYE instead of a REFER. This setting only applies when the next hop is not a session agent.
- SIP session agent—The SIP session agent takes precedence over the SIP interface. You add this ability to SIP session agents that need to receive a BYE instead of a REFER. If the next hop SIP element—the remote target in the dialog—is a session agent, in other words, you need to configure the option for it. Note that when you use this option for SIP session agents, the SIP interface or realm on which the REFER is received takes precedence over the REFER-to-BYE capability.

When a REFER request arrives and the REFER-to-BYE capability applies, the Oracle Enterprise Session Border Controller responds to it with a 202 Accepted and sends a NOTIFY to terminate the implicit refer subscription. This NOTIFY contains a message/sipfrag body with SIP/2.0 200 OK. Upon receiving the response to this NOTIFY, the Oracle Enterprise Session Border Controller sends a BYE with an added Reason header (encoded with the Refer-To number) to the other end.

The network element that does not accept REFERs takes the BYE with the Reason header and issues a new initial INVITE that initiates transfer, which the Oracle Enterprise Session Border Controller sees as starting a new and independent session.

SIP hold-refer-reinvite

When SIP hold-refer-reinvite is enabled for REFER with Replaces, the system queues the outgoing Invite populated from the received REFER based on the dialog state.

In a deployment where a call goes through the Oracle Enterprise Session Border Controller (E-SBC) before going to an Interactive Voice Response (IVR) server, the E-SBC proxies the intermediate reinvite that the IVR sends to the transfer target. If the intermediate reinvite is in either the pending state or the established state when the IVR initiates the transfer to the transfer target, the E-SBC terminates the call prematurely. The hold-refer-reinvite option allows the E-SBC to queue the Out Going INVITE from the received REFER request when the previously proxied reinvite request is in either the pending state or the established state. The result is a successful call.

Enable the SIP hold-refer-reinvite option from the ACLI command line or the Web GUI in Expert mode.

Enable hold-refer-reinvite - ACLI

The SIP hold-refer-reinvite parameter for REFER with Replaces is a parameter that you enable to prevent premature call termination in a deployment where calls are proxied by the Oracle Enterprise Session Border Controller.

- Confirm that refer-reinvite is added to realm/SA/SipInterface options.
- Confirm that refer-call-transfer is enabled on realm/SA/SipInterface
- Confirm that the session agent on which you want to enable hold-refer-reinvite is configured.

To enable hold-refer-reinvite, select a configured session agent and enable the parameter on the selected agent.

1. Access the **session-agent** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)
```

2. Type select, and press ENTER.
   The system displays a numbered list of session-agents.
3. Type the number of the agent on which you want to enable hold-refer-invite, and press ENTER.
4. Type hold-refer-reinvite enabled, and press ENTER.
5. Type done to save the configuration.

- Enable the refer-hold-reinvite parameter in the realm configuration.
- Enable the refer-hold-reinvite parameter in the session agent configuration.

SIP Roaming

This section explains how to configure SIP roaming. SIP roaming lets subscribers move from one active SIP device to another (at the same site or multiple sites) and retain service at the last registering device.

Overview

The Oracle Enterprise Session Border Controller supports multiple active registrations for the same user. The softswitch makes decisions regarding the current location of the user and the handling of requests from devices that are not currently identified as the user location. When there are multiple NATs, the Oracle Enterprise Session Border Controller is still required to let the softswitch be able to differentiate it.

The Oracle Enterprise Session Border Controller's SIP roaming ability supports the following features:

- Multiple active registrations from the same user can be cached, allowing subscribers to move from one active SIP device to another (at the same site or multiple sites) and still retain service at the last registering device. With the SIP roaming feature, one person, using multiple devices, can be contacted at all of the devices. These multiple devices (with their unique contact information) register to indicate that they are available for anyone that wants to contact that one person.
- The Oracle Enterprise Session Border Controller can also inform network devices (such as softswitches) of private SIP device IPv4 addresses (endpoints) and the public firewall address of the user location.

Process Overview

Caller 1 wants to contact Person A. Caller 1 sends a message to persona@acmepacket.com, but Person A has configured more than one SIP-enabled device to accept messages sent to that address. These devices have unique addresses of desk@10.0.0.4 and phone2@10.0.0.5. Person A has desk@10.0.0.4 and phone2@10.0.0.5 registered with the Oracle Enterprise Session Border Controller for anything addressed to persona@acmepacket.com.

With the SIP roaming feature, the Oracle Enterprise Session Border Controller accepts and stores both registrations for persona@acmepacket.com. That way, when someone wants to get in touch with Person A, the messages are sent to both devices (desk@10.0.0.4 and phone2@10.0.0.5) until Person A answers one of them. You do not need to

configure your Oracle Enterprise Session Border Controller for this functionality; your Oracle Enterprise Session Border Controller automatically provides it.

Using Private IPv4 Addresses

In addition to supporting multiple registries, the Oracle Enterprise Session Border Controller (E-SBC) can also distinguish user locations by their private IPv4 address and the IPv4 address of the public firewall. Using this information, the E-SBC adds private endpoint and public firewall information to Contact headers.

For example, entering this information causes a Contact header that formerly appeared as the following:

```
Contact:<sip:027411620263.67.143.217>
```

to subsequently appear as the following:

```
Contact:<sip:027411620263.67.143.217;ep=192.168.1.10;fw=10.1.10.21>
```

The E-SBC SIP proxy reads this information and populates the contact-endpoint and contact-firewall fields with the appropriate values.

Example 1 With a NAT Firewall

The Oracle Enterprise Session Border Controller (E-SBC) SIP proxy is configured with the following changeable parameters:

- endpoint= IP address of the SIP UA
- useradd= IP address of the Firewall Public IP address or the source layer 3 IP address of Register message
- userport= IP address port number of the Firewall Public IP address or the source layer 3 IP address port of Register message
- E-SBC address=63.67.143.217
- firewall public address=10.1.10.21
- firewall public address port=10000
- SIP endpoint behind firewall=192.168.1.10

SIP message Contact header:

```
Contact:<sip:027411620263.67.143.217; endpoint=192.168.1.10;
useradd=10.1.10.21; userport=10000; transport=udp>
```

Example 2 Without a NAT Firewall

The Oracle Enterprise Session Border Controller SIP proxy is configured with the following changeable parameters:

- useradd= IP address of the SIP UA or the source layer 3 IP address of Register message
- userport= IP address port number of the SIP UA or the source layer 3 IP address port of Register message
- Oracle Enterprise Session Border Controller address=63.67.143.217
- SIP endpoint=192.168.1.10
- SIP endpoint IP address port=5060

SIP message Contact header:

```
Contact:<sip:027411620263.67.143.217; useradd=192.168.1.10; userport=5060;
transport=udp>
```

For SIP, the softswitch responsibility is that the URI SD put in the Contact of the REGISTER message should be reflected in the 200-OK response to the REGISTER request. The Contact header of the response should have an expires header parameter indicating the lifetime of the registration.

The following example shows a Oracle Enterprise Session Border Controller Send:

```
Contact: <sep: 027411620263.67.143.217 endpoint=192.168.1.10;
useradd=10.1.10.21; userport=10000>;
```

The following examples shows the softswitch Respond:

```
Contact: <sep: 0274116202@63.67.143.217 endpoint=192.168.1.10;
useradd=10.1.10.21; userport=10000>;  expires=360
```

The contact field for endpoint and firewall parameters only appear in the following:

- Contact header of a REGISTER request sent from the Oracle Enterprise Session Border Controller to the softswitch server
- Contact header of a REGISTER response sent from the softwitch server to the Oracle Enterprise Session Border Controller
- Request-URI of an initial INVITE sent from the UT CSA server to the Oracle Enterprise Session Border Controller

An active endpoint is deleted when it does not register within the registration-interval setting or receives a 401 Unauthorized.

SIP Roaming Configuration

You can configure the SIP configuration's options parameter to indicate that you want to use the private IP address of the SIP device that the user is using and/or the public firewall address that identifies the location of the device. If defined, these options will be added as parameters to all Contact headers.

You can identify the endpoint and/or firewall information using the following options:

- contact-endpoint=<value> where <value> is the endpoint address or label
- contact-firewall=<value> where <value> is the firewall address or label

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.

4. Type options followed by a Space.
5. After the Space, type the information for an endpoint or a firewall, or both:

   ```
   contact-endpoint=”<label>
   contact-firewall=”<label>
   "contact-endpoint="<label>",contact-firewall="<label>""
   ```

6. Press Enter.

   For example, if you want your Oracle Enterprise Session Border Controller to add private endpoint and public firewall information to Contact headers, and you want to label this information as ep and fw, you would enter the following information in the ACLI.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)# options "contact-endpoint="ep",contact-
   firewall="fw""
   ```

Embedded Header Support

This section explains how to configure embedded header support. The Oracle Enterprise Session Border Controller supports methods of extracting an embedded P-Asserted-Identity header from a contact header to support E911 when

integrated with certain vendor's systems. See RFC 3455 *Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)* for more information.

The embedded header support feature watches for a specified embedded header contained in a Contact header received in a 3XX message. When the specified embedded header is found, the full <header=value> pair is inserted as a unique header in a redirected INVITE message that exits the Oracle Enterprise Session Border Controller. If the outgoing INVITE message were to contain the specified header, regardless of the use of this feature, the value extracted from the 3XX message replaces the INVITE message's specified header value.

If an incoming Contact header in a 3XX message looks like:

```
Contact: <ESRN@IPv4_Intrado_GW;user=phone?P-Asserted-Identity=%3Csip:+1-
ESQK@IPv4_My_EAG;user=phone%3E>
```

Then, if you configure your Oracle Enterprise Session Border Controller to parse for the embedded P-Asserted-Identity header to write as a unique header in the outgoing invite message, the outgoing INVITE and P-Asserted-Identity headers will look like:

```
INVITE SIP: ESRN@IPv4_Intrado_GW;user=phone
P-Asserted-Identity: +1-ESQK@IPv4_My_EAG;user=phone
```

Embedded Header Support Configuration

Embedded header support is enabled in the session agent configuration.

To configure embedded header support:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. Select the session agent where you want this feature.

   ```
   ACMEPACKET(session-agent)# select
   <hostname>:
   1: asd          realm=      ip=1.0.0.0
   2: SIPSA        realm=      ip=10.10.102.1
   selection:2
   ACMEPACKET(session-agent)#
   ```

5. request-uri-headers—Enter a list of embedded headers extracted from the Contact header that will be inserted in the re INVITE message. To configure this parameter for multiple headers, enclose the headers in double quotes and separate them with spaces. This completes the configuration of embedded header support.

   ```
   ACMEPACKET(session-agent)# request-uri-headers P-Asserted-Identity
   ```

Static SIP Header and Parameter Manipulation

This section explains the SIP header and parameter manipulation feature, which lets the Oracle Enterprise Session Border Controller add, modify, and delete SIP headers and parts of SIP headers called SIP header elements. SIP header elements are the different subparts of the header, such as the header value, header parameter, URI parameter and so on (excluding the header name).

To enable the SIP header and parameter manipulation functionality, you create header manipulation rulesets in which you specify header manipulation rules, as well as optional header element rules that operate on specified header elements. You then apply the header manipulation ruleset as inbound or outbound for a session agent or SIP interface.

Header Manipulation Rules

Header manipulation rules operate on the header you specify when you configure the rule. A header manipulation rule can also be configured with a list of element rules, each of which would specify the actions you want performed for a given element of this header.

Header Element Rules

Header element rules perform operations on the elements of a header. Header elements include all subparts of a header; excluding the header name. For example, header value, header parameter, URI parameter, and so on.

About SIP Header and Parameter Manipulation

Using the SIP header manipulation ruleset, you can cause the Oracle Enterprise Session Border Controller to:

- Delete a header based on header name match.
- Delete a header based on header name match as well as header value match.
- Add a header.
- Modify the elements of a header (by configuring header element rules):

  Add an element to a header.

  For example, add a parameter to a header or add a URI parameter to the URI in a header.

  Delete an element from a header.

  For example, delete a parameter from a header or delete a URI parameter from the URI in a header.

  Modify an element of a header.

  For example, replace a FQDN with an IPv4 address in a header or replace the value of a parameter in the header.

  Delete a message body part

  For example, delete the body part if the Content-Type is application/ISUP.

HMR $LOCAL_PORT for Port Mapping

When you configure SIP HMR and set an element-rule's new-value parameter to $LOCAL_PORT, the Oracle Enterprise Session Border Controller maps this value to the real port it uses for each signaling exchange.

Role in Trunk Group URI Feature

SIP header and parameter manipulation plays a role in the trunk group URI feature. You need to set the new-value parameter to one of the trunk group values when configuring SIP header rules, if using this feature. (In addition you can configure session agents and session agents groups on the Oracle Enterprise Session Border Controller to insert trunk group URI parameters in the SIP contact header.

For all trunk group URI support, you must set the appropriate parameters in the SIP header manipulation configuration and in the session agent or session agent group configurations.

For trunk group URI support, the SIP header and parameter manipulation configuration tells the Oracle Enterprise Session Border Controller where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

SIP Header and Parameter Manipulation Configuration

This section explains how to configure SIP header and parameter manipulation. First you create a SIP header manipulation ruleset, then the header manipulation rules and optional header element rules you want that ruleset to contain. You then configure a session agent or a SIP interface to use the SIP header and parameter manipulation ruleset in the inbound and outbound directions.

Creating SIP Header Manipulation Rulesets

To configure the SIP header manipulation ruleset:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-manipulation and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. name—Enter the name you want to use for this ruleset.
5. header-rules—Define the header manipulation rules you want to include in this ruleset.

   Type header-rules and press Enter.

```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```

   name—Enter the name of the header to which this rule applies. (The name you enter here must match a header name.)

   This is a case-insensitive string that is compared to the header name for matching. You need to create a rule using the long form of the header name and a rule using the compact form of the header name.

   Note: The Request-URI header is identified as request-uri.

   action—Enter the action you want applied to the header specified in the name parameter. The default value is none. Valid options are:

   - add—Add a new header, if that header does not already exist.
   - delete—Delete the header, if it exists.
   - manipulate—Elements of this header will be manipulated according to the element rules configured.
   - store—Store the header.
   - none—No action to be taken.

     match-value—Enter the value to be matched (only an exact match is supported) with a header value. The action you specify is only performed if the header value matches.

     msg-type—Enter the message type to which this header rule applies. The default value is any. Valid options are:

   - any—Both Requests and Reply messages
   - request—Request messages only
   - reply—Reply messages only

     Type show to display the header rule configuration values.
6. element-rules—Define the element rules you want to use to be performed on the elements of the header specified by the header rule.

   Type element-rules and press Enter.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

   name—Enter the name of the element to which this rule applies.

   Note: The name parameter usage depends on the element type you enter in step 6. For uri-param, uri-user-param, and header-param it is the parameter name to be added, replaced, or deleted. For all other types, it serves to identify the element rule and any name can be used.

type—Enter the type of element on which to perform the action. The default value is none. Valid options are:

- header-value—Enter value of the header.
- header-param-name—Header parameter name.
- header-param—Parameter portion of the header.
- uri-display—Display of the SIP URI.
- uri-user—User portion of the SIP URI.
- uri-host—Host portion of the SIP URI.
- uri-port—Port number portion of the SIP URI.
- uri-param-name—Name of the SIP URI param.
- uri-param—Parameter included in the SIP URI.
- uri-header-name—SIP URI header name
- uri-header—Header included in a request constructed from the URI.
- uri-user-param—User parameter of the SIP URI.

action—Enter the action you want applied to the element specified in the name parameter, if there is a match value. The default value is none. Valid options are:

- none—No action is taken.
- add—Add a new element, if it does not already exist.
- replace—Replace the elements.
- delete-element—Delete the specified element if it exists. Based on the match value if entered in step 6f.
- delete-header—Delete the specified header, if it exists.
- store—Store the elements.

match-val-type—Enter the type of value that needs to be matched to the match-field entry for the action to be performed. The default value is ANY. Valid options are:

- IP—Element value in the SIP message must be a valid IP address to be compared to the match-value field entry. If the match-value field is empty, any valid IP address is considered a match. If the element value is not a valid IP address, it is not considered a match.
- FQDN—Element value in the SIP message must be a valid FQDN to be compared to the match-value field entry. If the match-value field is empty, any valid FQDN is considered a match. If the element value is not a valid FQDN, it is not considered a match.
- ANY—Element value in the SIP message is compared with the match-value field entry. If the match-value field is empty, all values are considered a match.

match-value-—Enter the value you want to match against the element value for an action to be performed.

new-value-—Enter the value for a new element or to replace a value for an existing element. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators.

- Absolute values, with which you can use double quotes for clarity. You must escape all double quotes and back slashes that are part of an absolute value, and enclose the absolute value in double quotes.

  For example:

  sip:”+$TRUNK_GROUP+”.$TRUNK_GROUP_CONTEXT
- Pre-defined parameters always start with a $. Valid pre-defined parameters are:

Parameter	Description
$ORIGINAL	Original value of the element is used.
$LOCAL_IP	IP address of the SIP interface on which the message was received for inbound manipulation; or sent on for outbound manipulation.
$REMOTE_IP	IP address the message was received from for inbound manipulation; or being sent to for outbound manipulation.
$REMOTE_VIA_HOST	Host from the top Via header of the message is used.

Parameter	Description
$TRUNK_GROUP	Trunk group is used.
$TRUNK_GROUP_CONTEXT	Trunk group context is used.

- Operators are:

Operator	Description
+	Append the value to the end. For example: acme”+”packet generates acmepacket
+^	Prepends the value. For example: acme”+^”packet generates packetacme
-	Subtract at the end. For example: 112311”-”11 generates 1123
-^	Subtract at the beginning. For example: 112311”-^”11 generates 2311

Examples of entries for the new-value field.

```
$ORIGINAL+acme
$ORIGINAL+”my name is john”
$ORIGINAL+”my name is \”john\””
$ORIGINAL-^781+^617
```

Type show to display the element rule configuration values.

Type done to save them.

Repeat steps 6b through 6j to create additional rules.

Type exit to return to the header-rules parameters.

7. methods-—Enter the SIP method names to which you want to apply this header rule. If entering multiple method names, separate them with commas. For example:

```
INVITE,ACK,BYE
```

This field is empty by default. If you leave the method field empty, the header-rule is applied to all methods.

8. Type exit to return to the sip-manipulation level.
9. Save your work using the ACLI done command.
10. If you want to save this configuration, exit out of configuration mode and type save-config.

Configuring a Session Agent

You can configure a session agent to use the SIP header manipulation ruleset.

To configure a session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. in-manipulationid—Enter the name of the SIP header manipulation ruleset you want to apply to inbound SIP packets.

```
ACMEPACKET(session-agent)# in-manipulationid route-stripper
```

5. out-manipulationid—Enter the name of the SIP header manipulation ruleset you want to apply to outbound SIP packets.

```
ACMEPACKET(session-agent)# out-manipulationid route-stripper
```

6. Save your work using the ACLI done command.
7. If you want to save this configuration, exit out of configuration mode and type save-config.

Configuring a SIP Interface

You can configure a interface to use a SIP header manipulation ruleset.

To configure a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. in-manipulationid—Enter the name of the SIP header manipulation ruleset you want to apply to SIP packets in the ingress direction.

```
ACMEPACKET(sip-interface)# in-manipulationid
```

5. out-manipulationid—Enter the name of the SIP header manipulation ruleset you want to apply to SIP packets in the egress direction.

```
ACMEPACKET(sip-interface)# out-manipulationid
```

6. Save your work using the ACLI done command.
7. If you want to save this configuration, exit out of configuration mode and type save-config.

Example 1 Stripping All Route Headers

This example explains how to strip all route headers from a SIP packet. First, you create a header manipulation ruleset, in the example it is called route-stripper. Then you configure the list of header manipulation rules you need to strip route headers. In this case, you only need one rule named Route (to match the Route header name) with the action set to Delete.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)# name route-stripper
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)# name Route
ACMEPACKET(sip-header-rules)# action Delete
ACMEPACKET(sip-header-rules)# done
header-rule
```

```
        name                            Route
        action                          delete
        match-value
        msg-type                        any
ACMEPACKET(sip-header-rules)# ex
ACMEPACKET(sip-manipulation)# done
sip-manipulation
        name                            route-stripper
        header-rule
                name                            Route
                action                          delete
                match-value
                msg-type                        any
```

Example 2 Stripping an Existing Parameter and Adding a New One

This example explains how to strip the user parameter from the Contact header URI and add the acme parameter with value as LOCAL IP, only for requests. First you create a header manipulation ruleset, in the example it is called param-stripper1. You then configure a list of header rules you need. In this case, you only need one rule named Contact (to match the Contact header name), with action set to manipulate (indicating the elements of this header would be manipulated). Next, you configure a list of element rules for the Contact header rule.

In this case you configure two element rules; one to strip the uri parameter user (the rule name user matches the param name user) and the other to add the uri parameter acme (the rule name acme matches the param name acme).

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)# name param-stripper1
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)# name Contact
ACMEPACKET(sip-header-rules)# action manipulate
ACMEPACKET(sip-header-rules)# msg-type request
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)# name user
ACMEPACKET(sip-element-rules)# type uri-param
ACMEPACKET(sip-element-rules)# action delete-element
ACMEPACKET(sip-element-rules)# done
element-rule
        name                            user
        type                            uri-param
        action                          delete-element
        match-val-type                  any
        match-value
        new-value
ACMEPACKET(sip-element-rules)# name acme
ACMEPACKET(sip-element-rules)# action add
ACMEPACKET(sip-element-rules)# type uri-param
ACMEPACKET(sip-element-rules)# new-value "$LOCAL_IP"
ACMEPACKET(sip-element-rules)# done
element-rule
        name                            acme
        type                            uri-param
        action                          add
        match-val-type                  any
        match-value
        new-value                       "$LOCAL_IP"
ACMEPACKET(sip-element-rules)# ex
ACMEPACKET(sip-header-rules)# done
header-rule
        name                            Contact
        action                          manipulate
        match-value
        msg-type                        request
```

```
        element-rule
                name                            user
                type                            uri-param
                action                          delete-element
                match-val-type                  any
                match-value
                new-value
        element-rule
                name                            acme
                type                            uri-param
                action                          add
                match-val-type                  any
                match-value
                new-value                       "$LOCAL_IP"
ACMEPACKET(sip-header-rules)# ex
ACMEPACKET(sip-manipulation)# done
sip-manipulation
        name                            param-stripper1
        header-rule
                name                            Contact
                action                          manipulate
                match-value
                msg-type                        request
                element-rule
                        name                            user
                        type                            uri-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
element-rule
                name                            acme
                type                            uri-param
                action                          add
                match-val-type                  any
                match-value
                new-value                       "$LOCAL_IP"
```

For example, if the IP address of the SIP interface ($LOCAL_IP) is 10.1.2.3 and the Oracle Enterprise Session Border Controller receives the following Contact header:

```
Contact: <sip:1234@10.4.5.6;user=phone>
```

The header rule is applied to strip the user parameter from the Contact header URI and add the acme parameter with the value 10.1.2.3:

```
Contact: <sip:1234@10.4.5.6;acme=10.1.2.3>
```

SIP HMR (Header Manipulation Rules)

SIP header manipulation can also be configured in a way that makes it possible to manipulate the headers in SIP messages both statically and dynamically. Using this feature, you can edit response headers or the Request-URI in a request, and change the status code or reason phrase in SIP responses.

Static SIP Header and Parameter Manipulation allows you to set up rules in your Oracle Enterprise Session Border Controller configuration that remove and/or replace designated portions of specified SIP headers. SIP HMR allows you to set up dynamic header manipulation rules, meaning that the Oracle Enterprise Session Border Controller has complete control over alterations to the header value. More specifically:

- The Oracle Enterprise Session Border Controller can search header for dynamic content or patterns with the header value. It can search, for example, for all User parts of a URI that begin with 617 and end with 5555 (e.g., 617...5555).

- The Oracle Enterprise Session Border Controller can manipulate any part of a patterns match with any part of a SIP header. For example, 617 123 5555 can become 617 231 5555 or 508 123 0000, or any combination of those.

To provide dynamic header manipulation, the Oracle Enterprise Session Border Controller uses regular expressions to provide a high degree of flexibility for this feature. This allows you to search a specific URI when you do not know that value of the parameter, but want to use the matched parameter value as the header value. It also allows you to preserve matched sections of a pattern, and change what you want to change.

You can apply header manipulation to session agents, SIP interfaces, and realms. You do so by first setting up header manipulations rules, and then applying them in the configurations where they are needed. Within the header manipulation rules, there are sets of element rules that designate the actions that need to be performed on a given header.

Each header rule and each element rule (HMR) have a set of parameters that you configure to identify the header parts to be manipulated, and in what way the Oracle Enterprise Session Border Controller is to manipulate them. These parameters are explained in detail, but the parameter that can take regular expression values is match-value. This is where you set groupings that you want to store, match against, and manipulate.

Generally, you set a header rule that will store what you want to match, and then you create subsequent rules that operate on this stored value. Because header rules and element rules are applied sequentially, it is key to note that a given rule performs its operations on the results of all the rules that you have entered before it. For example, if you want to delete a portion of a SIP header, you would create Rule 1 that stores the value for the purpose of matching, and then create Rule 2 that would delete the portion of the header you want removed. This prevents removing data that might be used in the other header rules.

Given that you are using regular expression in this type of configuration, this tightly sequential application of rules means that you must be aware of the results to be yielded from the application of the regular expressions you enter. When you set a regular expression match value for the first rule that you enter, the Oracle Enterprise Session Border Controller takes the results of that match, and then a second rule might exist that tells the Oracle Enterprise Session Border Controller to use a new value if it the second rule's match value finds a hit (and only 10 matches, 0-9, are permitted) for the results (yield) from applying the first rule.

Consider the example of the following regular expression entry made for a match-value parameter:'Trunk(.+)', which might be set as that match value in the first rule you configure. Given a SIP element rule called uri-param and the param-name tgid, it can yield two values:

- Grouping 0—The entire matching string (Trunk1)
- Grouping 1—The grouping (1)

In turn, these groupings can be referenced in an element rule by using this syntax:

```
$<header rule name >.$<element rule name.$<value>
```

Additional syntax options that can be used with this feature are:

- $headerName['['index']']
- $headerName['['index']'][.$index]
- $headerName['['index']'][.$elementName]
- $headerName['['index']'][.$elementName][.$index]

Guidelines for Header and Element Rules

Header rules and element rules share these guidelines:

- References to groupings that do not exist result in an empty string.
- References to element rule names alone result in a Boolean condition of whether the expression matched or not.
- A maximum of ten matches are allowed for a regular expression. Match 0 (grouping 0) is always the match of the entire matching string; subsequent numbers are the results for other groups that match.

Precedence

The Oracle Enterprise Session Border Controller applies SIP header rules in the order you have entered them. This guards against the Oracle Enterprise Session Border Controller removing data that might be used in the other header rules.

This ordering also provides you with ways to use manipulations strategically. For example, you might want to use two rules if you want to store the values of a regular expression. The first rule would store the value of a matched regular expression, and the second could delete the matched value.

In addition to taking note of the order in which header rules are configured, you now must also configure a given header rule prior to referencing it. For example, you must create Rule1 with the action store for the Contact header BEFORE you can create Rule2 which uses the stored value from the Contact header.

Duplicate Header Names

If more than one header exists for the header name you have configured, the Oracle Enterprise Session Border Controller stores the value where it can be referenced with the optional syntax $<header rule name>[index]. Additional stored header values are indexed in the order in which they appear within the SIP message, and there is no limit to the index.

Possible index values are:

- ~ — The Oracle Enterprise Session Border Controller references the first matching header
- * — The Oracle Enterprise Session Border Controller references all headers
- ^ — The Oracle Enterprise Session Border Controller references the last stored header in the header rule

Performing HMR on a Specific Header

HMR has been enhanced so that you can now operate on a specific instance of a given header. The syntax you use to accomplish this is similar to that you used to refer to a specific header rule stored value instance.

Using the header-name parameter, you can now add a trailing [<index>] value after the header name. This [<index>] is a numerical value representing the specific instance of the header on which to operate. However, the Oracle Enterprise Session Border Controller takes no action if the header does not exist. You can also use the caret (^) to reference the last header of that type (if there are multiple instances)

The count for referencing is zero-based, meaning that the first instance of the header counts as 0.

Note that the header instance functionality has no impact on HMR's add action, and you cannot use this feature to insert headers into a specific location. Headers are added to the end of the list, except that Via headers are added to the top.

Multiple SIP HMR Sets

In general you use SIP HMR by configuring rules and then applying those rules to session agents, realms, or SIP interfaces in the inbound or outbound direction. In addition, the Oracle Enterprise Session Border Controller has a set method for how certain manipulation rules take precedence over others. For instance, inbound SIP manipulation rules defined in a session agent take precedence over any configured for a realm, and the rules for a realm take precedence over SIP interface manipulation rules.

The multiple SIP HMR feature gives you the ability to:

- Apply multiple inbound and outbound manipulations rules to a SIP message
- Provision the order in which the Oracle Enterprise Session Border Controller applies manipulation rules

The action parameter in the header rules configuration now takes the value sip-manip. When you set the parameter to sip-manip, you then configure the new-value parameter with the name of a SIP manipulation rule that you want to invoke. The values for the match-value, comparison-type, and methods parameters for invoked rule are all supported. This means that the manipulation defined by the rules identified in the new-value parameter are carried out when the values for the match-value, comparison-type, and methods parameters are true.

The relationship between manipulation rules and manipulation rule sets is created once you load your configuration, meaning that the order in which you enter them does not matter. It also means that the Oracle Enterprise Session Border Controller cannot dynamically perform validation as you enter rules, so you should use the ACLI verify-config command to confirm your manipulation rules contain neither invalid nor circular references. Invalid references are those that point to SIP manipulation rules that do not exist, and circular references are those that create endless loops of manipulation rules being carried out over and over. If you load a configuration exhibiting either of these issues, the Oracle Enterprise Session Border Controller forces the action value for the rule to none and the rule will not be used.

MIME Support

Using the SIP HMR feature set, you can manipulate MIME types in SIP message bodies. While you can manipulate the body of SIP messages or a specific content type using other iterations of SIP HMR, this version gives you the power to change the MIME attachment of a specific type within the body by using regular expressions. To achieve this, you use the find-replace-all action type, which enables the search for a particular string and the replacement of all matches for that type. Although you use find-replace-all to manipulate MIME attachments, it can also be used to achieve other goals in SIP HMR.

Note that using find-replace-all might consume more system resources than other HMR types. Therefore this powerful action type should only be used when another type cannot perform the type of manipulation you require.

Find and Replace All

To manipulate a particular portion of the MIME attachment, for example when removing a certain attribute within the content type of application/sdp, the Oracle Enterprise Session Border Controller (E-SBC) would need to search the content multiple times because:

- SDP can have more than one media line
- The SIP message body can contain more than one application/sdp.

The find-replace-all action type works for SIP header rules and for element rules. You can use it for all manipulation types from the entire header value, to the URI specific parameters, to MIME attachment.

For this action type, it does not matter what you configure the comparison type, which is atypical for actions types, as the comparison type is vital to the others. Find-replace-all, however, binds the comparison type to the pattern rule. Thus, the E-SBC treats the match value as a regular expression, and it ignores any configured comparison type value in favor of the pattern rule. This type of action is both a comparison and action: For each regular expression match within the supplied string, the E-SBC substitutes the new value for that match. Yet if you want to replace a certain portion of the regular expression and not the entire matched expression, you need to use a subgroup of expressions and the right syntax to indicate the sub-group replacement index.

You can indicate the sub-group replacement syntax by adding the string [[:n:]] to the end of the regular expression—where n is a number between 0 and 9. For example, given the following settings:

- action=find-replace-all
- match-value=sip:(user)@host[[:1:]]
- new-value=bob

you create a new rule to replace only the user portion of the URI that searches for the regular expression and replaces all instances of the user subgroup with the value bob.

Taking advantage of the find-replace-all's recursive nature, you can replace all the 0 digits in a telephone number with 1:

- action=find-replace-all
- match-value=0
- new-value=1

So for the user portion of a URI—or for any other string—with a value 1-781-308-4400 would be replaced as 1-781-318-4411.

If you leave the new-value parameter blank for find-replace-all, the E-SBC replaces the matched sub-group with an empty string—an equivalent of deleting the sub-group match. You can also replace empty sub-groups, which is like inserting a value within the second sub-group match. For example, user()@host.com[[:1:]]with a configured new-value _bob yields user_bob@host.com.

When you use find-replace-all, you cannot use the following parameter-type values: uri-param-name, uri-header-name, and header-param-name. These values are unusable because the E-SBC only uses case-sensitive matches for the match-value to find the parameter name within the URI. Since it can only be found by exact match, the E-SBC does not support finding and replacing that parameter.

Escaped Characters

SIP HMR's support for escaped characters allows for searches for values you would be unable to enter yourself. Because they are necessary to MIME manipulation, support for escaped characters now includes:

- \f
- \n
- \r
- \t
- \v

New Reserved Word

To allow you to search for carriage returns an new lines, the SIP HMR MIME feature also adds support for the reserved word $CRLF. Because you can search for these value and replace them, you also must be able to add them back in when necessary. Configuring $CRLF in the new-value parameter always resolves to /r/n, which you normally cannot otherwise enter through the ACLI.

About the MIME Value Type

Introduced to modify the MIME attachment, SIP HMR supports a mime value for the type parameter in the element rules configuration. Like the status-code and reason-phrase values, you can only use the mime type value against a specific header—which in this case, is Content-Type.

When you set the element rule type to mime, you must also configure the parameter-name with a value. This step is a requirement because it sets the content-type the Oracle Enterprise Session Border Controller manipulates in a specific part of the MIME attachment. You cannot leave this parameter blank; the Oracle Enterprise Session Border Controller does not let you save the configuration if you do. When you use the store action on a multi-part MIME attachment that has different attachment types, the Oracle Enterprise Session Border Controller stores the final instance of the content-type because it does not support storing multiple instances of element rule stored values.

In the event you do not know the specific content-type where the Oracle Enterprise Session Border Controller will find the match-value, you can wildcard the parameter-name by setting with the asterisk (*) as a value. You cannot, however, set partial content-types (i.e., application/*). So configured, the Oracle Enterprise Session Border Controller loops through the MIME attachment's content types.

You can set the additional action types listed in this table with the described result:

Action Type	Description
delete-element	Removes the matched mime-type from the body. If this is the last mime-type within in message body, the Oracle Enterprise Session Border Controller removes the Content-Type header.
delete-header	Removes all body content and removes the Content-Type header.
replace	Performs a complete replacement of the matched mime-type with the new-value you configure.

Action Type	Description
find-replace-all	Searches the specifies mime-type's contents and replaces all matching regular expressions with the new-value you configure
store	Stores the final instance of the content-type (if there are multi-part MIME attachments of various attachment types)
add	Not supported

MIME manipulation does not support manipulating headers in the individual MIME attachments. For example, the Oracle Enterprise Session Border Controller cannot modify the Content-Type given a portion of a message body like this one:

```
--boundary-1
Content-Type: application/sdp
v=0
o=use1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 10000 RTP/AVP 8
a=rtpmap:8 PCMA/8000/1
a=sendrecv
a=ptime:20
a=maxptime:200
```

Back Reference Syntax

You can use back reference syntax in the new-value parameter for header and element rules configurations. Denoted by the use of $1, $2, $3, etc. (where the number refers to the regular expression's stored value), you can reference the header and header rule's stored value without having to use the header rule's name. It instead refers to the stored value of this rule.

For example, when these settings are in place:

- header-rule=changeHeader
- action=manipulate
- match-value=(.+)([^;])

you can set the new-value as sip:$2 instead of sip:$changeHeader.$2.

You can use the back reference syntax for:

- Header rule actions manipulate and find-replace-all
- Element rule actions replace and find-replace-all

Using back reference syntax simplifies your configuration steps because you do not need to create a store rule and then manipulate rule; the manipulate rule itself performs the store action if the comparison-type is set to pattern-rule.

Notes on the Regular Expression Library

In the regular expression library, the dot (.) character no longer matches new lines or carriage returns. Conversely, the not-dot does match new lines and carriage returns. This change provides a safety mechanism preventing egregious backtracking of the entire SIP message body when there are no matches. Thus, the Oracle Enterprise Session Border Controller reduces backtracking to a single line within the body. In addition, there is now support for:

Syntax	Description
\s	Whitespace
\S	Non-whitespace

Syntax	Description
\d	Digits
\D	Non-digits
\R	Any \r, \n, \r\n
\w	Word
\W	Non-word
\A	Beginning of buffer
\Z	End of buffer
\	Any character including newline, in the event that the dot (.) is not

In addition, there is:

- Escaped character shortcuts (\w\W\S\s\d\D\R) operating inside brackets [...]

SIP Message-Body Separator Normalization

The Oracle Enterprise Session Border Controller supports SIP with Multipurpose Internet Mail Extension (MIME) attachments — up to a maximum payload size of 64KB — and has the ability to allow more than the required two CRLFs between the SIP message headers and the multipart body's first boundary. The first two CRLFs that appear in all SIP messages signify the end of the SIP header and the separation of the header and body of the message, respectively. Sometimes additional extraneous CRLFs can appear within the preamble before any text.

The Oracle Enterprise Session Border Controller works by forwarding received SIP messages regardless of whether they contain two or more CRLFs. Although three or more CRLFs are legal, some SIP devices do not accept more than two.

The solution to ensuring all SIP devices accept messages sent from the Oracle Enterprise Session Border Controller is to strip all CRLFs located at the beginning of the preamble before the appearance of any text, ensuring that there are no more than two CRLFs between the end of the last header and the beginning of the body within a SIP message. You enable this feature by adding the new stripPreambleCrlf option to the global SIP configuration.

To enable the stripping of CRLFs in the preamble:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET#(configure)
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET#(session-router)
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name stripPreambleCrlf with a plus sign.

```
ACMEPACKET(sip-config)# options +stripPreambleCrlf
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the global SIP configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

SIP Header Pre-Processing HMR

By default, the Oracle Enterprise Session Border Controller (E-SBC) performs in-bound SIP manipulations after it carries out header validation. Adding the inmanip-before-validate option in the global SIP configuration allows the E-SBC to perform HMR on received requests prior to header validation. Because there are occaisional issues with other SIP implementations—causing invalid headers to be used in messages they send to the E-SBC—it can be beneficial to use HMR to remove or repair these faulty headers before the request bearing them are rejected.

When configured to do so, the E-SBC performs pre-validation header manipulation immediately after it executes the top via check. Inbound SIP manipulations are performed in order of increasing precedence: SIP interface, realm, and session agent.

The fact that the top via check happens right before the E-SBC carries out pre-validation header manipulations means that you cannot use this capability to repairs the first via header if it is indeed invalid. If pre-validation header manipulation were to take place at another time during processing, it would not be possible to use it for SIP session agents. The system learns of matching session agents after top via checking completes.

For logistical reasons, this capability does not extend to SIP responses. Inbound manipulation for responses cannot be performed any sooner that it does by default, a time already preceding any header validation.

To enable SIP header pre-processing:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET#(configure)
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET#(session-router)
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name inmanip-before-validate with a plus sign.

```
ACMEPACKET(sip-config)# options +inmanip-before-validate
```

   This value allows a the E-SBC to perform pre-validation header manipulation in order of increasing precedence: SIP interface, realm, and session agent.

5. Save and activate the configuration.

Best Practices

This section lists practices that AOracle recommends you follow for successful implementation of this feature.

- Define all storage rules first.

  This recommendation is made because each subsequent header rule processes against the same SIP message, so each additional header rules works off of the results from the application of the rule that precedes it.

  In general, you want to store values from the original SIP header rather than from the iteratively changed versions.
- Implement rules at the element rule rather than the header rule level.

  Header rules should only be a container for element rules.
- When you are creating rules to edit a header, add additional element rules to modify a single header rather than try to create multiple header rules each with one element rule. That is, create multiple element rules within a header rule rather than creating multiple header rules.
- Do not use header or element rule names that are all capital letters (i.e., $IP_ADDRESS). Capitals currently refer to predefined rules that are used as macros, and they might conflict with a name that uses capital letters.

About Regular Expressions

Two of the most fundamental ideas you need to know in order to work with regular expressions and with this feature are:

- Regular expressions are a way of creating strings to match other string values.
- You can use groupings in order to create stored values on which you can then operate.

To learn more about regex, you can visit the following Web site, which has information and tutorials that can help to get you started:*http://www.regular-expressions.info/*.

Many of the characters you can type on your keyboard are literal, ordinary characters—they present their actual value in the pattern. Some characters have special meaning, however, and they instruct the regex function (or engine which interprets the expressions) to treat the characters in designated ways. The following table outlines these “special characters” or metacharacters.

Character	Name	Description
.	dot	Matches any one character, including a space; it will match one character, but there must be one character to match. Literally a . (dot) when bracketed ([]), or placed next to a \ (backslash).
*	star/asterisk	Matches one or more preceding character (0, 1, or any number), bracketed carrier class, or group in parentheses. Used for quantification. Typically used with a . (dot) in the format .* to indicate that a match for any character, 0 or more times. Literally an * (asterisk) when bracketed ([]).
+	plus	Matches one or more of the preceding character, bracketed carrier class, or group in parentheses. Used for quantification. Literally a + (plus sign) when bracketed ([]).
|	bar/vertical bar/pipe	Matches anything to the left or to the right; the bar separates the alternatives. Both sides are not always tried; if the left does not match, only then is the right attempted. Used for alternation.
{	left brace	Begins an interval range, ended with } (right brace) to match; identifies how many times the previous singles character or group in parentheses must repeat. Interval ranges are entered as minimum and maximums ({minimum,maximum}) where the character/group must appear a minimum of times up to the maximum. You can also use these character to set magnitude, or exactly the number of times a character must appear; you can set this, for example, as the minimum value without the maximum ({minimum,}).
?	question mark	Signifies that the preceding character or group in parentheses is optional; the character or group can appear not at all or one time.

Character	Name	Description
^	caret	Acts as an anchor to represent the beginning of a string.
$	dollar sign	Acts as an anchor to represent the end of a string.
[	left bracket	Acts as the start of a bracketed character class, ended with the] (right bracket). A character class is a list of character options; one and only on of the characters in the bracketed class must appear for a match. A - (dash) in between two character enclosed by brackets designates a range; for example [a-z] is the character range of the lower case twenty-six letters of the alphabet. Note that the] (right bracket) ends a bracketed character class unless it sits directly next to the [(left bracket) or the ^ (caret); in those two cases, it is the literal character.
(	left parenthesis	Creates a grouping when used with the) (right parenthesis). Groupings have two functions: They separate pattern strings so that a whole string can have special characters within it as if it were a single character. They allow the designated pattern to be stored and referenced later (so that other operations can be performed on it).

Expression Building Using Parentheses

You can now use parentheses (())when you use HMR to support order of operations and to simplify header manipulation rules that might otherwise prove complex. This means that expressions such as (sip + urp) - (u + rp) can now be evaluated to sip. Previously, the same expression would have evaluated to sipurprp. In addition, you previously would have been required to create several different manipulation rules to perform the same expression.

SIP Manipulation Configuration

This section explains the parameters that appear in the subelements for the SIP manipulations configuration. Within the SIP manipulations configuration, you can set up SIP header rules, and within those header rules you can configure element rules.

This section also contains several configuration examples for different applications of the HMR feature.

Configuring SIP Header Manipulation Rules

To configure dynamic SIP header manipulation rules:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter.

```
ACMEPACKET(sip-manipulation)# header-rules
```

5. name—Enter the unique identifier for this SIP HMR. There is no default for this value.
6. header-name—Enter the name of the header on which you want the Oracle Enterprise Session Border Controller (E-SBC) to use this HMR. There is no default for this parameter.

   Set this parameter to @status-line, where the at-sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name status-code.
7. msg-type—Specify the type of message to which this SIP HMR will be applied. The default value is any. The valid values are:

   - any | request | reply
8. methods—Enter the method type to use when this SIP HMR is used, such as INVITE, ACK, or CANCEL. When you do not set the method, the E-SBC applies the rule across all SIP methods.
9. comparison-type—Enter the way that you want SIP headers to be compared from one of the available. This choice dictates how the E-SBC processes the match rules against the SIP header. the default is refer-case-sensitive. The valid values are:.

   - boolean | refer-case-sensitive | refer-case-insensitive | pattern-rule | case-sensitive | case-insensitive
10. action—Enter the action that you want this rule to perform on the SIP header. The default value is none. The valid values are:

    - add | delete | manipulate | store | none

      Remember that you should enter rules with the action type store before you enter rules with other types of actions.

      When you set the action type to store, the E-SBC always treats the match value you enter as a regular expression. As a default, the regular expression is uses for the match value is .+ (which indicates a match value of at least one character), unless you set a more specific regular expression match value.
11. match-value—Enter the value to match against the header value in SIP packets; the E-SBC matches these against the entire SIP header value. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators.

    When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the match-value parameter to support escaping Boolean and string manipulation operators..

    You can also escape the escape character itself, so that it is used as a literal string. For example, the E-SBC now treats the string \+1234 as +1234.

    The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “.

    You can also use two variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
12. new-value—When the action parameter is set to add or to manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the E-SBC to use when it adds or manipulates SIP headers.

    When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the new-value parameter to support escaping Boolean and string manipulation operators..

    You can also escape the escape character itself, so that it is used as a literal string. For example, the E-SBC now treats the string \+1234 as +1234.

    The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “.

    You can also use two variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

Configuring SIP Header Manipulation Element Rules

Element rules are a subset of the SIP header manipulation rules and are applied at the element type level rather than at the entire header value.

To configure dynamic SIP header manipulation rules:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter.

```
ACMEPACKET(sip-manipulation)# header-rules
```

5. Type element-rules and press Enter.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

6. name—Enter the unique identifier for this element rule. There is no default for this value.
7. parameter-name—Enter the SIP header parameter/element on which you want the Oracle Enterprise Session Border Controller (E-SBC) to use this rule. There is no default for this parameter.
8. type—Specify the type of parameter to which this element rule will be applied. The default value is none. The valid values are:

   - header-value | header-param-name | header-param | uri-display | uri-user | uri-user-param | uri-host | uri-port | uri-param-name | uri-param | uri-header-name | uri-header

     To configure HMR so that there is impact only on the status-line; the value will be used for matching according to the comparison-type:
   - status-code—Designates the status code of the response line; accepts any string, but during the manipulation process only recognizes the range from 1 to 699.
   - reason-phrase—Designates the reason of the response line; accepts any string.
9. match-val-type—Enter the value type that you want to match when this rule is applied. The default value is ANY. Valid values are:

   - IP | FQDN | ANY
10. comparison-type—Enter the way that you want SIP headers to be compared from one of the available. This choice dictates how the E-SBC processes the match rules against the SIP header parameter/element. The default is refer-case-sensitive.

    - boolean | refer-case-sensitive | refer-case-insensitive | pattern-rule
11. action—Enter the action that you want this rule to perform on the SIP header parameter/element. The default is none. The valid rules are:

    - add | replace | delete-element | delete-header | store | none

      Remember that you should enter rules with the action type store before you enter rules with other types of actions.

      When you set the action type to store, the E-SBC always treats the match value you enter as a regular expression. As a default, the regular expression is uses for the match value is .+ (which indicates a match value of at least one character), unless you set a more specific regular expression match value.

12. match-value—Enter the value to match against the header value in SIP packets; the E-SBC matches these against the value of the parameter/element. This is where you can enter values to match using regular expression values, or stored pattern matches. Your entries can contain Boolean operators.

    When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the match-value parameter to support escaping Boolean and string manipulation operators..

    You can also escape the escape character itself, so that it is used as a literal string. For example, the E-SBC now treats the string \+1234 as +1234.

    The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “.

    You can also use two variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

13. new-value—When the action parameter is set to add or to manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the E-SBC to use when it adds or manipulates parameters/elements.

    When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the new-value parameter to support escaping Boolean and string manipulation operators..

    You can also escape the escape character itself, so that it is used as a literal string. For example, the E-SBC now treats the string \+1234 as +1234.

    The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “.

    You can also use two variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

Status-Line Manipulation and Value Matching

The Oracle Enterprise Session Border Controller’s HMR feature has been enhanced to support the ability to change the status code or reason phrase in SIP responses. This addition—the ability to edit status-lines in responses—builds on HMR’s existing ability to edit response headers or the Request-URI in a request.

This section shows you how to configure SIP HMR when you want the Oracle Enterprise Session Border Controller to drop a 183 Session Progress response when it does not have SDP, though flexibility is built into this feature so that you can use it to achieve other ends. In addition, you can now set the SIP manipulation’s match-value parameter with Boolean parameters (AND or OR).

Setting the Header Name

SIP header rules (part of the SIP manipulation configuration) now support a new value for the header-name parameter. The value is @status-line, where the at-sign (@)—not allowed in SIP header names—prevents undesired matches with header having the name status-code.

To set the header name for SIP header rules:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. header-name—Enter the new value for the header-name parameter: @status-line.

Setting the Element Type

In the element rules (a subset of the SIP header rules configuration), you can now set the type parameter to either of the following values, both of which will only have an impact on the status-line:

- status-code—Designates the status code of the response line; accepts any string, but during the manipulation process only recognizes the range from 1 to 699
- reason-phrase—Designates the reason of the response line; accepts any string

Like other rule types you can set, the Oracle Enterprise Session Border Controller matches against the value for these using case-sensitive, case-insensitive, or pattern-rule matching (set in the comparison-type parameter for the element rule).

To set the element type:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. Type element-rule and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

6. type—Enter either status-code or reason-phrase, the value of which will be used for matching according to the comparison-type.

Setting the Match Value

Note that for the SIP header rules and for the SIP element rules, the match-value parameter can now be set with these Boolean operators:

- and (for which the syntax is the ampersand &)
- or (for which the syntax is the pipe|)

However, you can only use Boolean operators in this value when you set the comparison-type parameter to pattern-rule and are evaluating stored matches. The Oracle Enterprise Session Border Controller evaluates these Boolean expressions from left to right, and does not support any grouping mechanisms that might change the order of evaluation. For example, the Oracle Enterprise Session Border Controller evaluates the expression A & B | C (where A=true, B=false, and C=true) as follows: A & B = false; false | true = true.

You can set the match-value for the SIP header rules or for the SIP element rules.

To set a match value in the SIP header rules configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. match-value—Enter the value to match against the header value in SIP packets; the Oracle Enterprise Session Border Controller matches these against the entire SIP header value. This is where you can enter values to match using regular expression values; your entries can contain Boolean operators.

To set a match value in the SIP element rules configuration:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

7. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

9. Type header-rules and press Enter.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```

10. Type element-rule and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

11. match-value—Enter the value to match against the header value in SIP packets; the Oracle Enterprise Session Border Controller matches these against the value of the parameter/element. This is where you can enter values to match using regular expression values, or stored pattern matches; your entries can contain Boolean operators.

Setting the Response Code Block

To enable the SIP HMR enhancements, you need to set an option in SIP interface configuration that keeps the Oracle Enterprise Session Border Controller from sending the response you designate.

Note that this example sets the dropResponse option to 699, where 699 is an arbitrary code used to later match the HMR.

To enable SIP response blocking for a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI select command) the configuration that you want to edit.

4. options—Set the options parameter by typing options, a Space, the option name dropResponse with a plus sign in front of it, type the equal sign and the code(s) or range(s) you want blocked. If there is more than one, separate your entries with a colon. Then press Enter.

```
ACMEPACKET(sip-interface)# options +dropResponse=699
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Configuring MIME Support

The find-replace-all action has been added to the header rules. Element rules support the find-replace-all action and the mime type.

To set the header rule with the find-replace-all action:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter.

```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. action—Set the action parameter to find-replace-all if you want to enable SIP HMR MIME manipulation.
6. Save and activate your configuration.

   To set the element rule with the find-replace-all action and MIME type:

7. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

8. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

9. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

10. Type header-rules and press Enter.

```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```

11. Type element-rules and press Enter.

```
ACMEPACKET(sip-header-rules)# element-rules
```

12. ACMEPACKET(sip-element-rules)#
13. action—Set the action parameter to find-replace-all if you want to enable SIP HMR MIME manipulation.
14. type—Set the type parameter to mime if you want to enable SIP HMR MIME manipulation.
15. Save and activate your configuration.

Testing Pattern Rules

The Oracle Enterprise Session Border Controller it yields the results you require. This command is useful for testing the regex values that you devise because it will tell you whether that value is valid or not.

This command is called test-pattern-rule.

To test a pattern rule:

1. Type test-pattern-rule and press Enter.

```
ACMEPACKET# test-pattern-rule
ACMEPACKET(test-pattern-rule)#
```

2. expression—Enter the regular expression that you want to test. If there is a match, then the Oracle Enterprise Session Border Controller will inform you of it; you will also be informed if there is no match.

   The string against which the Oracle Enterprise Session Border Controller is matching is not the string parameter that you can use for this command; it is the string value of the regular expression you entered.

```
ACMEPACKET(test-pattern-rule)# expression '.*;tgid=(.+).*'
```

3. string—Enter the string against which you want to compare the regular expression.

```
ACMEPACKET(test-pattern-rule)# string sip:+17024260002@KCMGGWC;user=phone
SIP/2.0;tgid=Trunk1
expression made 3 matches against string
```

4. show—Use the show command within test-pattern-rules to view the test pattern that you entered, whether there was a match, and the number of matches.

```
ACMEPACKET(test-pattern-rule)# show
Pattern Rule:
Expression : .*(;tgid=(.+)).*
String     : sip:+17024260002@KCMGGWC;user=phone SIP/2.0;tgid=Trunk1
Matched    : TRUE
Matches:
$0 sip:+17024260002@KCMGGWC;user=phone SIP/2.0;tgid=Trunk1
$1 ;tgid=Trunk1
$2 Trunk1
```

Configuring SIP HMR Sets

This section shows you how to configure your multiple SIP HMR sets.

Remember to run the ACLI verify-config command prior to activating your configuration so the Oracle Enterprise Session Border Controller can detect any invalid or circular references.

To set the parameters enabling the use of SIP HMR sets:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type header-rules and press Enter.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. Type element-rule and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

6. action—Enter the sip-manip value to enable use this rule for a SIP HMR set. This value then invoke the rule identified in the new-value parameter.
7. new-value—To use SIP HMR sets, enter the name of the manipulation rule you want invoked for the set.
8. Save and activate your configuration.

Configuration Examples

This section shows you several configuration examples for HMR. This section shows the configuration for the various rules that the Oracle Enterprise Session Border Controllerapplied, and sample results of the manipulation. These examples present configurations as an entire list of fields and settings for each ruleset, nested header rules and nested element rules. If a field does not have any operation within the set, the field is shown with the setting at the default or blank.

Example 1 Removing Headers

For this manipulation rule, the Oracle Enterprise Session Border Controller removes the Custom header if it matches the pattern rule. It stores the defined pattern rule for the goodBye header. Finally, it removes the goodBye header if the pattern rule from above is a match.

This is a sample of the configuration:

```
sip-manipulation
        name                            removeHeader
        header-rule
                name                            removeCustom
                header-name                     Custom
                action                          delete
                comparison-type                 boolean
                match-value                     ^This is my.*
                msg-type                        request
                new-value
                methods                         INVITE
        header-rule
                name                            goodByeHeader
                header-name                     Goodbye
                action                          store
comparison-type     boolean
                match-value                     ^Remove (.+)
                msg-type                        request
                new-value
                methods                         INVITE
header-rule
                name                            goodBye
action                          delete
                comparison-type                 pattern-rule
                match-value                     $goodByeHeader
                msg-type                        request
                new-value
                methods                         INVITE
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
```

```
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 2 Manipulating the Request URI

For this manipulation rules, the Oracle Enterprise Session Border Controller stores the URI parameter tgid in the Request URI. Then if the pattern rule matches, it adds a new header (x-customer-profile) with the a new header value tgid to the URI parameter in the request URI.

This is a sample of the configuration:

```
sip-manipulation
        name                            CustomerTgid
        header-rule
                name                            ruriRegex
                header-name                     request-uri
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
new-value
                methods                         INVITE
                element-rule
                        name                            tgidParam
                        parameter-name                  tgid
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
header-rule
                name                            addCustomer
                header-name                     X-Customer-Profile
                action                          add
                comparison-type                 pattern-rule
                match-value                     $ruriRegex.$tgidParam
                msg-type                        request
                new-value                       $ruriRegex.$tgidParam.$0
                methods                         INVITE
header-rule
                name                            delTgid
                header-name                     request-uri
                action                          manipulate
                comparison-type                 pattern-rule
                match-value                     $ruriRegex.$tgidParam
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            tgidParam
                        parameter-name                  tgid
                        type                            uri-param
                        action                          delete-element
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     $ruriRegex.$tgidParam.
```

```
$0
                        new-value
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060 SIP/2.0
    Message Header
Via: SIP/2.0/UDP 192.168.200.61:5060;branch=z9hG4bK0g6plv3088h03acgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rg601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDc1rg601-f125d8b0ec7985c378b04cab9f91cc09-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
        X-Customer-Profile: 123
```

Example 3 Manipulating a Header

For this manipulation rule, the Oracle Enterprise Session Border Controllerstores the pattern matches for the Custom header, and replaces the value of the Custom header with a combination of the stored matches and new content.

This is a sample of the configuration:

```
sip-manipulation
        name                            modCustomHdr
        header-rule
                name                            customSearch
                header-name                     Custom
                action                          store
                comparison-type                 pattern-rule
                match-value                     (This is my )(.+)( header)
                msg-type                        request
                new-value
                methods                         INVITE
header-rule
                name                            customMod
                header-name                     Custom
                action                          manipulate
                comparison-type                 pattern-rule
                match-value                     $customSearch
                msg-type                        request
                new-value
methods                         INVITE
                element-rule
                        name                            hdrVal
                        parameter-name                  hdrVal
                        type                            header-value
                        action                          replace
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
new-value                       $customSearch.$1+edited+$customSearch.$3
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
```

```
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK20q2s820boghbacgs6o0.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDe1ra601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDe1ra601-4bb668e7ec9eeb92c783c78fd5b26586-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my edited header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 4 Storing and Using URI Parameters

For this manipulation rule, the Oracle Enterprise Session Border Controller stores the value of the URI parameter tag from the From header. It also creates a new header FromTag with the header value from the stored information resulting from the first rule.

This is a sample of the configuration:

```
sip-manipulation
        name                            storeElemParam
        header-rule
                name                            Frohmr
                header-name                     From
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            elementRule
                        parameter-name                  tag
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value
header-rule
                name                            newHeader
                header-name                     FromTag
                action                          add
                comparison-type                 pattern-rule
                match-value                     $FromHR.$elementRule
                msg-type                        any
                new-value                       $FromHR.$elementRule.$0
                methods
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK4oda2e2050ih7acgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDf1re601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDf1re601-f85059e74e1b443499587dd2dee504c2-06a3gu0
```

```
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
Content-Length: 140
        FromTag: 1
```

Example 5 Manipulating Display Names

For this manipulation rule, the Oracle Enterprise Session Border Controller sores the display name from the Display header. It replaces the two middle characters of the original display name with a new string. Then is also replaces the From header's display name with "abc 123" if it matches sipp.

This is a sample of the configuration:

```
sip-manipulation
        name                            modDisplayParam
        header-rule
                name                            storeDisplay
                header-name                     Display
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            displayName
                        parameter-name                  display
                        type                            uri-display
                        action                          store
                        match-val-type                  any
comparison-type                 pattern-rule
                        match-value                     (s)(ip)(p )
                        new-value
header-rule
                name                            modDisplay
                header-name                     Display
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            modRule
                        parameter-name                  display
                        type                            uri-display
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $storeDisplay.
$displayName
                        new-value                       $storeDisplay.
$displayName.$1+lur+$storeDisplay.$displayName.$3
header-rule
                name                            modFrom
```

```
                header-name                     From
                action                          manipulate
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            fromDisplay
                        parameter-name
                        type                            uri-display
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     sipp
                        new-value                       "\"abc 123\" "
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681kot109gp04acgs6o0.1
        From: "abc 123" <sip:sipp@192.168.1.60:5060>;tag=SD79ra601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD79ra601-a487f1259e2370d3dbb558c742d3f8c4-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: slurp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 6 Manipulating Element Parameters

For this more complex manipulation rule, the Oracle Enterprise Session Border Controller:

- From the Display header, stores the display name, user name, URI parameter up, and header parameter hp
- Adds the header parameter display to the Params header, with the stored value of the display name from the first step
- Add the URI parameter user to the Params header, with the stored value of the display name from the first step
- If the URI parameter match succeeds in the first step, replaces the URI parameter up with the Display header with the value def
- If the header parameter match succeeds in the first step, deletes the header parameter hp from the Display header

This is a sample of the configuration:

```
sip-manipulation
        name                            elemParams
        header-rule
                name                            StoreDisplay
                header-name                     Display
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
```

```
                element-rule
                        name                            displayName
                        parameter-name
                        type                            uri-display
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            userName
                        parameter-name                  user
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            uriParam
                        parameter-name                  up
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            headerParam
                        parameter-name                  hp
                        type                            header-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
        header-rule
                name                            EditParams
                header-name                     Params
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            addHeaderParam
                        parameter-name                  display
                        type                            header-param
                        action                          add
match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $StoreDisplay.
$displayName.$0
                element-rule
                        name                            addUriParam
                        parameter-name                  user
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
              new-value
```

```
$StoreDisplay.$userName.$0
        header-rule
                name                            EditDisplay
                header-name                     Display
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            replaceUriParam
                        parameter-name                  up
                        type                            uri-param
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $StoreDisplay.$uriParam
                        new-value                       def
                element-rule
                        name                            delHeaderParam
                        parameter-name                  hp
                        type                            header-param
                        action                          delete-element
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value $StoreDisplay.$headerParam
                        new-value
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK7okvei0028jgdacgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD89rm601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD89rm601-b5b746cef19d0154cb1f342cb04ec3cb-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=def>
        Params: sipp <sip:sipp1@192.168.1.60:5060;user=user>;display=sipp
        Params: sipp <sip:sipp2@192.168.1.60:5060;user=user>;display=sipp
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 7 Accessing Data from Multiple Headers of the Same Type

For this manipulation rule, the Oracle Enterprise Session Border Controller stores the user name from the Params header. It then adds the URI parameter c1 with the value stored from the first Params header. Finally, it adds the URI parameter c2 with the value stored from the second Params header.

This is a sample of the configuration:

```
sip-manipulation
        name                            Params
        header-rule
                name                            storeParams
                header-name                     Params
                action                          store
```

```
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            storeUserName
                        parameter-name                  user
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value
header-rule
                name                            modEdit
                header-name                     Edit
                action                          manipulate
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
                new-value
methods                         INVITE
                element-rule
                        name                            addParam1
                        parameter-name                  c1
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $storeParams[0].
$storeUserName.$0
                element-rule
                        name                            addParam2
                        parameter-name                  c2
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $storeParams[1].
$storeUserName.$0
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK9g855p30cos08acgs6o0.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD99ri601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD99ri601-6f5691f6461356f607b0737e4039caec-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060;c1=sipp1;c2=sipp2>
        Max-Forwards: 69
        Subject: Performance Test
```

```
        Content-Type: application/sdp
        Content-Length: 140
```

Example 8 Using Header Rule Special Characters

For this manipulation rule, the Oracle Enterprise Session Border Controller:

- Stores the header value of the Params header with the given pattern rule, and stores both the user name of the Params header and the URI parameter abc
- Adds the URI parameter lpu with the value stored from the previous Params header
- If any of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value aup
- If all of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value apu
- If the first Params headers does not match the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter not with the value 123
- If the first Params headers matches the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter yes with the value 456

This is a sample of the configuration:

```
sip-manipulation
        name                            specialChar
        header-rule
                name                                    searchParams
                header-name                             Params
                action                                  store
                comparison-type                         pattern-rule
                match-value                             .*sip:(.+)@.*
                msg-type                                request
                new-value
                methods                                 INVITE
                element-rule
                        name                                    userName
                        parameter-name
                        type                                    uri-user
                        action                                  store
                        match-val-type                          any
                        comparison-type                         case-sensitive
                        match-value
                        new-value
element-rule
                        name                                    emptyUriParam
                        parameter-name                          abc
                        type                                    uri-param
                        action                                  store
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value
                        new-value
header-rule
                name                                    addUserLast
                header-name                             Edit
                action                                  manipulate
                comparison-type                         case-sensitive
                match-value
                msg-type                                request
                new-value
                methods                                 INVITE
                element-rule
                        name                                    lastParamUser
                        parameter-name                          lpu
                        type                                    uri-param
```

```
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value $searchParams[^].$userName.$0
                element-rule
                        name                            anyParamUser
                        parameter-name                  apu
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $searchParams[~]
                        new-value                       aup
                element-rule
                        name                            allParamUser
                        parameter-name                  apu
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $searchParams[*]
                        new-value                       apu
                element-rule
                        name                            notParamYes
                        parameter-name                  not
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     !$searchParams.
$emptyUriParam
                        new-value                       123
                element-rule
                        name                            notParamNo
                        parameter-name                  yes
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $searchParams.
$emptyUriParam
                        new-value                       456
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681m9t30e0qh6akgj2s1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDchrc601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDchrc601-fcf5660a56e2131fd27f12fcbd169fe8-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp
<sip:user@192.168.1.60:5060;lpu=sipp2;apu=aup;not=123>;apu=apu
        Max-Forwards: 69
        Subject: Performance Test
```

```
        Content-Type: application/sdp
        Content-Length: 140
```

Example 9 Status-Line Manipulation

This section shows an HMR configuration set up for status-line manipulation.

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the Oracle Enterprise Session Border Controller searches the 183 Session Progress responses where the content length is zero
3. If the 183 Session Progress response does not contain SDP, change its status code to 699
4. Drop all 699 responses

```
sip-manipulation
        name                            manip
        description
        header-rule
                name                            IsContentLength0
                header-name                     Content-Length
                action                          store
                comparison-type                 pattern-rule
                match-value                     0
                msg-type                        reply
                new-value
                methods
        header-rule
                name                            is183
                header-name                     @status-line
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
name                            is183Code
                        parameter-name
                        type                            status-code
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     183
                        new-value
        header-rule
                name                            change183
                header-name                     @status-line
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
                        name                            make199
                        parameter-name
                        type                            status-code
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $IsContentLength0 &
$is183.$is183Code
```

```
                        new-value                       199

sip-interface     options dropResponse=699
```

Example 10 Use of SIP HMR Sets

The following example shows the configuration for SIP HMR with one SIP manipulation configuration loading another SIP manipulation configuration. The goals of this configuration are to:

- Add a new header to an INVITE
- Store the user portion of the Request URI
- Remove all Route headers from the message only if the Request URI is from a specific user

```
sip-manipulation
        name                            deleteRoute
        description                     delete all Route Headers
        header-rule
                name                            deleteRoute
                header-name                     Route
                action                          delete
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
sip-manipulation
        name                            addAndDelete
        description                     Add a New header and delete Route
headers
        header-rule
                name                            addHeader
                header-name                     New
                action                          add
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value                       "Some Value"
                methods                         INVITE
        header-rule
                name                            storeRURI
                header-name                     request-uri
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            storeUser
                        parameter-name
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     305.*
                        new-value
        header-rule
                name                            deleteHeader
                header-name                     request-uri
                action                          sip-manip
                comparison-type                 Boolean
                match-value                     $storeRURI.$storeUser
                msg-type                        request
```

```
                new-value                       deleteRoute
                methods                         INVITE
```

Example 11 Use of Remote and Local Port Information

The following example shows the configuration for remote and local port information. The goals of this configuration are to:

- Add LOCAL_PORT as a header parameter to the From header
- Add REMOTE_PORT as a header parameter to the From header

```
sip-manipulation
        name                            addOrigIp
        description
        header-rule
                name                            addIpParam
                header-name                     From
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            addIpParam
                        parameter-name                  newParam
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $LOCAL_IP
                element-rule
                        name                            addLocalPort
                        parameter-name                  lport
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $LOCAL_PORT
                element-rule
                        name                            addRemotePort
                        parameter-name                  rport
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $REMOTE_PORT
```

Example 12 Response Status Processing

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the Oracle Enterprise Session Border Controller searches the 183 Session Progress responses where the content length is zero
3. If the 183 Session Progress response does not contain SDP, change its status code to 699
4. Drop all 699 responses

```
sip-manipulation
        name                            manip
```

```
        description
        header-rule
                name                            IsContentLength0
                header-name                     Content-Length
                action                          store
                comparison-type                 pattern-rule
                match-value                     0
                msg-type                        reply
                new-value
                methods
        header-rule
                name                            is183
                header-name                     @status-line
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
                        name                            is183Code
                        parameter-name
                        type                            status-code
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     183
                        new-value
        header-rule
                name                            change183
                header-name                     @status-line
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
                        name                            make699
                        parameter-name
                        type                            status-code
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $IsContentLength0 &
$is183.$is183Code
                        new-value                       699
sip-interface
    options dropResponse=699
```

The following four configuration examples are based on the this sample SIP INVITE:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
```

```
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap:8 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=ptime:20
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
a=ptime:30
--boundary--
```

Example 13 Remove a Line from SDP

In this example, the SIP manipulation is configured to remove all p-time attributes from the SDP.

```
sip-manipulation
        name                            removePtimeFromBody
        description                   removes ptime attribute from all bodies
        header-rule
                name                            CTypeManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            remPtime
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=ptime:[0-9]{1,2}(\n|
\r\n)
                        new-value
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
```

```
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
--boundary-
```

Example 14 Back Reference Syntax

In this sample of back-reference syntax use, the goal is to change the To user. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            changeToUser
        description                   change user in the To header
        header-rule
                name                            ChangeHeader
                header-name                     To
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            replaceValue
                        parameter-name
                        type                            header-value
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     (.+)(service)(.+)
                        new-value                       $1+Bob+$3
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:Bob@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
…
…
…
```

Example 15 Change and Remove Lines from SDP

In this sample of changing and removing lines from the SDP, the goal is to convert the G.729 codec to G.729a. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            std2prop-codec-name
        description                     rule to translate standard to
proprietary codec name
        header-rule
                name                            CTypeManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        any
                new-value
                methods
                element-rule
                        name                            g729-annexb-no-std2prop
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=rtpmap:[0-9]{1,3}
(G729/8000/1\r\na=fmtp:[0-9]{1,3} annexb=no)[[:1:]]
                        new-value                       G729a/8000/1
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 8
a=rtpmap:18 G729a/8000/1
a=sendrecv
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
--boundary-
```

Example 16 Change and Add New Lines to the SDP

In this sample of changing and adding lines from the SDP, the goal is to convert non-standard codec H.263a to H.263. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            prop2std-codec-name
        description                     rule to translate proprietary to
standard codec name
        header-rule
                name                            CodecManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        any
                new-value
                methods
                element-rule
                        name                            H263a-prop2std
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=rtpmap:([0-9]{1,3})
H263a/.*\r\n
                        new-value                       a=rtpmap:+$1+"
H263/90000"+$CRLF+a=fmtp:+$1+" QCIF=4"+$CRLF
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 8
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263/90000
a=fmtp:34 QCIF=4
--boundary-
```

Dialog-Matching Header Manipulation

The most common headers to manipulate using HMR are the To-URI and From-URI. Along with the to-tag, from-tag, and Call-ID values, these are also all headers that represent dialog-specific information that must match the UAC and UAS to be considered part of the same dialog. If these parameters are modified through HMR, the results can be that the UAC or UAS rejects messages.

While it is possible to ensure that dialog parameters match correctly using regular HMR, this feature offers a simpler and less error-prone method of doing so.

In addition, this section describes the addition of built-in SIP manipulations defined by Oracle best practices, and a new method of testing your SIP manipulations.

About Dialog-Matching Header Manipulations

The goal of this feature is to maintain proper dialog-matching through manipulation of dialog-specific information using HMR. Two fundamental challenges arise when looking at the issue of correctly parameters manipulating dialog-matching:

- Inbound HMR
- Outbound HMR

The new setting out-of-dialog (for the msg-type parameter) addresses these challenges by offering an intelligent more of dialog matching of messages for inbound and outbound HMR requests. This is a msg-type parameter, meaning that it becomes matching criteria for operations performed against a message. If you also specify methods (such as REGISTER) as matching criteria, then the rule is further limited to the designated method.

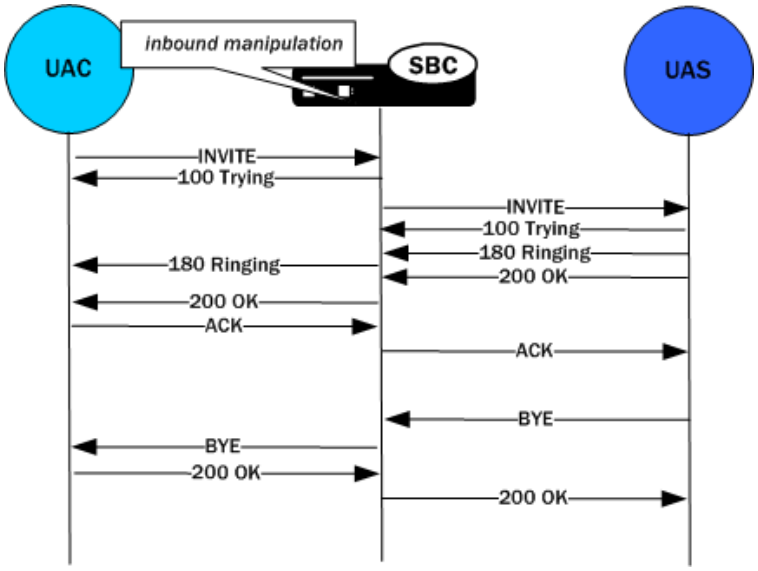
For both inbound and outbound manipulations, using the out-of-dialog setting means the message must be a request without a to-tag in order to perform the manipulation.

Inbound HMR Challenge

Since inbound manipulations take place before the message reaches the core of Oracle Enterprise Session Border Controller SIP processing, the SIP proxy takes the manipulated header as what was directly received from the client. This can cause problems for requests leaving the Oracle Enterprise Session Border Controller for the UAC because the dialog will not match the initial request sent.

So the unmodified header must be cached because for any subsequent request (as in the case of a BYE originating from the terminator; see the diagram below) the Oracle Enterprise Session Border Controller might need to restore the original value—enabling the UAC to identify the message correctly as being part of the same dialog. For out-of-dialog requests (when the To, From, or Call-ID headers are modified) the original header will be stored in the dialog when the msg-type out-of-dialog is used.

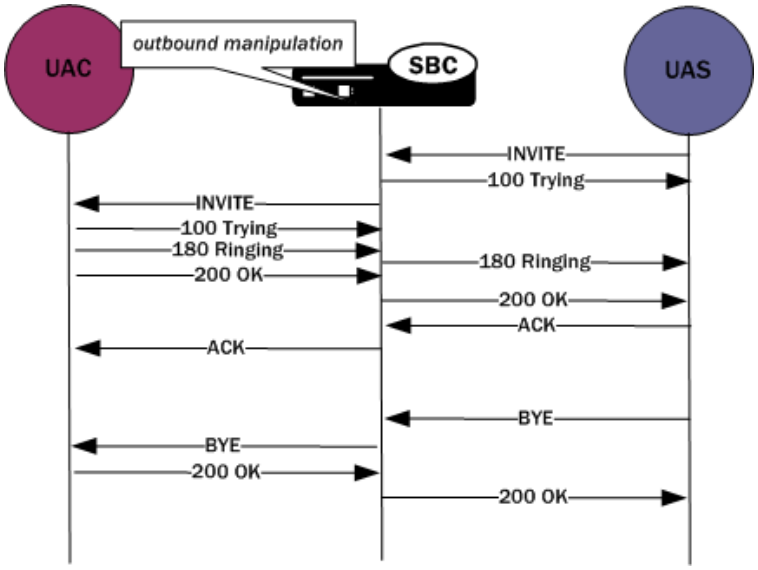
The Oracle Enterprise Session Border Controller performs the restoration of original headers outside of SIP manipulations. That is, there are no manipulation rules to configure for restore the header to their original context. The Oracle Enterprise Session Border Controller will recognize the headers have been modified, and restore them to their original state prior to sending the message out on the wire. Restoration takes place prior to outbound manipulations so that any outbound manipulation can those headers once they have been restored.



Outbound HMR Challenge

When you use the out-of-dialog setting for an outbound manipulation, the Oracle Enterprise Session Border Controller only executes this specific SIP header rule only if the same SIP header rule was executed against the initial dialog-creating request.

For example, if the INVITE's To header was not manipulated, it would not be correct to manipulate the To header in the BYE request. To do so would render the UAC unable to properly match the dialog. And this also means that the outbound manipulation should be carried out against a To, From, or Call-ID header in the BYE request if it was manipulated in the INVITE.



Dialog-matching Header Manipulation Configuration

You using the out-of-dialog setting in the msg-type parameter, part of the SIP header rules configuration.

To enable dialog-matching header manipulation:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-manipulation and press Enter.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type mime-rules and press Enter. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```

5. msg-type—Set this parameter to out-of-dialog to enable dialog-matching header manipulation.
6. Save your work.

Built-In SIP Manipulations

In the course of HMR use, certain rules have become commonly used. Lengthy and complex, these rules do not include any customer-specific information and do they can be used widely. To make using them easier, they have been turned into built-in rules that you can reference in the in-manipulationid and out-manipulationid parameters that are part of the realm, session agent, and SIP interfaces configurations.

Built-in rules start with the prefix ACME_, so Oracle recommends you name your own rules in a different manner to avoid conflict.

While the number of built-in manipulation rules is expected to grow, one is supported at the present time: ACME_NAT_TO_FROM_IP. When performed outbound, this rule changes:

- The To-URI hostname to the logical $TARGET_IP and port to $TARGET_PORT
- The From-URI to the logical $REPLY_IP and port to be $REPLY_PORT

Built-In SIP Manipulation Configuration

When you want to enable this feature for a realm, session agent, or SIP interface, you configure the in-manipulationid or out-manipulationid parameters with the rule.

The sample here shows this feature being applied to a session agent, but the realm and SIP interface configurations also have the same parameter you use to set up the feature.

To use built-in SIP manipulations:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. out-manipulationid—Enter name of the built-in rule you want to use. Remember that all built-in rules start with ACME_.
5. Save your work.

Testing SIP Manipulations

You can now use a new tool that allows you to test the outcome of your SIP manipulation and header rules without sending real traffic through the Oracle Enterprise Session Border Controller to see if they work.

To use the tool, you enter the ACLI's test-sip-manipulation utility and reference the rule you want to test using this name. Then you enter a mode where you put in a SIP message entered in ASCII. You can cut and paste this message from sipmsg.log or from some other location. Using <Ctrl-D> stops the SIP message collection and parses it.

The test informs you of any parsing errors found in the SIP message. Once the message is entered, you can execute the SIP manipulation against the message. The output after this step is the modified SIP message after manipulations have been applied. You will also find a debugging option, which displays SIP manipulation logging to the screen as the manipulation takes place.

As a starting point for testing, this tool comes loaded with a default SIP message. It cannot be associated with realms, session agents, or SIP interfaces, and so it also comes with certain resolves reserved words, such as: $LOCAL_IP, $TRUNK_GROUP_CONTEXT, and $REMOTE_PORT. In addition, you can use your settings for testing across terminal sessions; if you choose to save your settings, everything (including the SIP message) will be saved, with the exception of the debugging option.

It is not recommended that you use this tool to add an ISUP message body.

HMR Import-Export

Due to the complexity of SIP manipulations rules and the deep understanding of system syntax they require, it is often difficult to configure reliable rules. This feature provides support for importing and exporting pieces of SIP manipulation configuration in a reliable way so that they can be reused.

Exporting

The SIP manipulation configuration contains an export command. When you use it, the Oracle Enterprise Session Border Controller sends the configuration you have selected to a designated file. The contents are the same information you see when you use the ACLI show command in XML format; it includes the selected configuration and any changes that have been made. Because you can only export one SIP manipulation configuration at a time, you must export each one-by-one if you need more than one.

The file name can be any you selected, and would be most useful if it were to identify its contents in some way. If the file already exists, then the export fails and informs you the file already exists. A successfully-executed export simply returns you to the system prompt.

The system writes exported files to /code/imports, a new location that will be created to avoid overlap with existing backup files. The files will carry the extension .gz to show that they have been compressed with gzip.

Your export data will look like this sample:

```
<?xml version='1.0' standalone='yes'?>
<sipManipulation
        name='manip'
        description=''
        lastModifiedBy='admin@console'
        lastModifiedDate='2009-10-16 14:16:29'>
        <headerRule
                headerName='Foo'
                msgType='any'
                name='headerRule'
                action='manipulate'
                cmpType='boolean'
                matchValue='$REGEX("[bB][A-Za-z]{2}")'
                newValue='foo'
                methods='INVITE'>
        </headerRule>
</sipManipulation>
```

To avoid conflict with other objects on the system, key and object ID are not included as part of the exported XML.

Importing

Using the import command in the SIP manipulation configuration, you can import data from an exported file to a currently-selected configuration. If you have not selected a configuration into which to load the data, a new one will be created. Including the .gz extension, you enter the full name of the file you want imported. After it finds the file, the Oracle Enterprise Session Border Controller unarchives it and parses its contents. If these steps fail, the Oracle Enterprise Session Border Controller will alert you. If they succeed, then the configuration data loads into the object.

If you have been making changes to the configuration into which data was imported, the Oracle Enterprise Session Border Controller will inform you prior to importing the data so that you will not lose any of your work. This way, you will be less likely to overwrite unsaved changes.

Once the import is complete, it will be as if you entered the configuration by hand. You only need to save your work (by typing done) to save the SIP manipulation to the global SIP configuration. Note that if for some reason the XML is malformed or contained more than one object, the import will fail.

If you attempt to import a configuration with the same key as one that already exists, the system returns an error and prevents you from saving the imported object. In this case, you can delete the object with the same key and then carry out your import, or you can select the object with the same key and perform an import that will overwrite it with new data.

Displaying Imports

You can display imported SIP manipulations data at the system prompt. The command lists all files in the exported files directory, and also tells you if there are none.

Using FTP to Move Files

You can also place exported SIP manipulation configuration files on the Oracle Enterprise Session Border Controller using FTP. You need to use the same /code/imports directory to do so.

Removing Files

Using the delete-import command with the name of the file you want to delete removes it from the system. Using this command, you can delete files that are no longer useful to you. Carrying out this command is final and there is no warning before you go ahead with the deletion. A failed deletion (for instance, because there is no such file) will produce an error message; a successful deletion simply returns you to the system prompt.

Unique HMR Regex Patterns and Other Changes

In addition to the HMR support it offers, the Oracle Enterprise Session Border Controller can now be provisioned with unique regex patterns for each logical remote entity. This supplement to pre-existing HMR functionality saves you provisioning time and saves Oracle Enterprise Session Border Controller resources in instances when it was previously necessary to define a unique SIP manipulation per PBX for a small number of customer-specific rules.

Manipulation Pattern Per Remote Entity

On the Oracle Enterprise Session Border Controller, you can configure logical remote entities (session agents, realms, and SIP interfaces) with a manipulation pattern string that the system uses as a regular expression. Then the SIP manipulation references this regular expression using the reserved word $MANIP_PATTERN. At runtime, the Oracle Enterprise Session Border Controller looks for the logical entity configured with a manipulation pattern string in this order of preference: session agent, realm, and finally SIP interface.

On finding the logical entity configured with the manipulation string, the Oracle Enterprise Session Border Controller dynamically determines the expression. When there is an invalid reference to a manipulation pattern, the pattern-rule expression that results will turn out to be the default expression (which is \,+).

When the $MANIP_PATTERN is used in a manipulation rule's new-value parameter, it resolves to an empty string, equivalent of no value. Even though this process ends with no value, it still consumes system resources. And so Oraclerecommends you do not use $MANIP_PATTERN as a new-value value.

In the following example, the SIP manipulation references the regular expression from a realm configuration:

```
realm-config
        identifier                              net200
        description
        addr-prefix                             0.0.0.0
        network-interfaces                      public:0
        ...
        manipulation-pattern                    Lorem(.+)
sip-manipulation
        name                            manip
        description
        header-rules
                name                            headerRule
                header-name                     Subject
                action                          manipulate
                match-value                     $MANIP_PATTERN
                msg-type                        request
                comparison-type                 pattern-rule
                new-value                       Math
                methods                         INVITE
```

Reject Action

When you use this action type and a condition matching the manipulation rule arises, the Oracle Enterprise Session Border Controller rejects the request (though does not drop responses) and increments a counter.

- If the msg-type parameter is set to any and the message is a response, the Oracle Enterprise Session Border Controller increments a counter to show the intention to reject the message—but the message will continue to be processed.
- If the msg-type parameter is set to any and the message is a request, the Oracle Enterprise Session Border Controller performs the rejection and increments the counter.

The new-value parameter is designed to supply the status code and reason phrase corresponding to the reject. You can use the following syntax to supply this information: status-code[:reason-phrase]. You do not have to supply the status code and reason phrase information; by default, the system uses 400:Bad Request.

If you do supply this information, then the status code must be a positive integer between 300 and 699. The Oracle Enterprise Session Border Controller then provides the reason phrase corresponding to the status code. And if there is no reason phrase, the system uses the one for the applicable reason class.

You can also customize a reason phrase. To do so, you enter the status code followed by a colon (:), being sure to enclose the entire entry in quotation marks () if your reason code includes spaces.

When the Oracle Enterprise Session Border Controller performs the reject action, the current SIP manipulation stops processing and does not act on any of the rules following the reject rule. This course of action is true for nested SIP manipulations that might have been constructed using the sip-manip action type.

Reject Action Configuration

To support the reject action, two parameters in the session-router-config allow you to set how many messages in a certain amount of time cause the Oracle Enterprise Session Border Controller to generate an SNMP trap.

To set the reject message number and time window:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-router and press Enter.

```
ACMEPACKET(session-router)# session-router
ACMEPACKET(session-router-config)#
```

4. reject-message-threshold—Enter the minimum number of message rejections allowed in the reject-message-window time on the Oracle Enterprise Session Border Controller (when using the SIP manipulation action reject) before generating an SNMP trap. The default is 0, meaning this feature is disabled and no trap will be sent.
5. reject-message-window—Enter the time in seconds that defines the window for maximum message rejections allowed before generating an SNMP trap.
6. Save your work.

About Counters

The Oracle Enterprise Session Border Controller tracks messages that have been flagged for rejection using the reject action type. In the show sipd display, refer to the Rejected Messages category; there is no distinction between requests and responses.

```
ACMEPACKET# show sipd
13:59:07-102
SIP Status                -- Period -- -------- Lifetime --------
                Active    High   Total       Total  PerMax    High
Sessions             0       0       0           0       0       0
Subscriptions        0       0       0           0       0       0
Dialogs              0       0       0           0       0       0
CallID Map           0       0       0           0       0       0
Rejections           -       -       0           0       0
ReINVITEs            -       -       0           0       0
Media Sessions       0       0       0           0       0       0
Media Pending        0       0       0           0       0       0
Client Trans         0       0       0           0       0       0
Server Trans         0       0       0           0       0       0
Resp Contexts        0       0       0           0       0       0
Saved Contexts       0       0       0           0       0       0
Sockets              0       0       0           0       0       0
Req Dropped          -       -       0           0       0
DNS Trans            0       0       0           0       0       0
DNS Sockets          0       0       0           0       0       0
DNS Results          0       0       0           0       0       0
Rejected Msgs        0       0       0           0       0       0
Session Rate = 0.0
Load Rate = 0.0
Remaining Connections = 20000 (max 20000)
```

SNMP Support

The Oracle Enterprise Session Border Controller provides SNMP support for the Rejected Messages data, so you can access this information externally. The new MIB objects are:

```
apSysRejectedMessages    OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
                "Number of messages rejected by the SD due to matching
criteria."
        ::= { apSysMgmtMIBGeneralObjects 18 }
apSysMgmtRejectedMesagesThresholdExeededTrap        NOTIFICATION-TYPE
        OBJECTS         { apSysRejectedMessages }
        STATUS          current
        DESCRIPTION
        " The trap will be generated when the number of rejected messages
```

```
exceed the configured threshold within the configured window."
        ::= { apSystemManagementMonitors 57 }
apSysMgmtRejectedMessagesGroup  OBJECT-GROUP
        OBJECTS {
            apSysRejectedMessages
        }
        STATUS          current
        DESCRIPTION
                "Objects to track the number of messages rejected by the SD."
        ::= {  apSystemManagementGroups 18 }
apSysMgmtRejectedMessagesNotificationsGroup NOTIFICATION-GROUP
      NOTIFICATIONS {
                        apSysMgmtRejectedMesagesThresholdExeededTrap
                      }
                STATUS          current
                DESCRIPTION
                "Traps used for notification of rejected messages"
      ::= {  apSystemManagementNotificationsGroups 26 }
apSmgmtRejectedMessagesCap
            AGENT-CAPABILITIES
            PRODUCT-RELEASE     "Acme Packet SD"
            STATUS               current
            DESCRIPTION         "Acme Packet Agent Capability for enterprise
                                 system management MIB."
            SUPPORTS            APSYSMGMT-MIB
                INCLUDES        {
                                  apSysMgmtRejectedMessagesGroup,
                                  apSysMgmtRejectedMessagesNotificationsGroup
                                }
            ::= { apSmgmtMibCapabilities 37 }
```

Log Action

When you use this action type and a condition matching the manipulation rule arises, the Oracle Enterprise Session Border Controller logs information about the current message to a separate log file. This log files will be located on the same core in which the SIP manipulation occurred. On the core where sipt runs, a logfile called matched.log will appear when this action type is executed.

The matched.log file contains a timestamp, received and sent Oracle Enterprise Session Border Controller network interface, sent or received IP address:port information, and the peer IP address:port information. It also specifies the rule that triggered the log action in this syntax: rule-type[rule:name]. The request URI, Contact header, To Header, and From header are also present.

```
---------------------------------------
Apr 17 14:17:54.526 On [0:0]192.168.1.84:5060 sent to 192.168.1.60:5060
element-rule[checkRURIPort]
INVITE sip:service@192.168.1.84:5060 SIP/2.0
From: sipp <sip:+2125551212@192.168.1.60:5060>;tag=3035SIPpTag001
To: sut <sip:service@192.168.1.84>
Contact: sip:sipp@192.168.1.60:5060
```

Changes to Storing Pattern Rule Values

Release S-C6.2.0 introduces changes to the framework for storing regular expression results within manipulation rules, altering the way the store action works. These changes are beneficial to performance.

In previous releases, when the store action is used, the Oracle Enterprise Session Border Controller stores all values matching the regular expression defined in the match-value parameter for all headers. At runtime, the system evaluates all stored values to find the correct index.

Now, you no longer need to specify the store action. The simple fact of referencing another rule tells the system it must store a value. When SIP manipulation is used, the system first checks to see if any values require storing. The add action is an exception to this process; storing happens after a header is added.

When referring to a rule, that rule still needs to have a regular expression defined in the match-vale and the comparison type set to pattern-rule; else the default expression will be used.

Removal of Restrictions

The following restrictions related to HMR have been removed in Release S-C6.2.0:

- The action find-replace-all now executes all element rules. Previously, no child rules were executed.
- The action sip-manip now executes existing all element rules. Previously, no child rules were executed.
- The action store now executes existing all element rules. Previously, only child rules with the store action were executed.
- The action add now executes existing all element rules. Previously, only child rules with the add action were executed.

Name Restrictions for Manipulation Rules

Historically, you have been allowed to configure any value for the name parameter within a manipulation rule. This method of naming caused confusion when referencing rules, so now manipulation rules name must follow a specific syntax. They must match the expression ^[[alpha:]][[:alnum:]_]+$ and contain at least one lower case letter.

In other words, the name must:

- Start with a letter, and then it can contain any number of letters, numbers, or underscores
- Contain at least one lower case letter

All pre-existing configurations will continue to function normally. If you want to change a manipulation rule, however, you are required to change its name if it does not follow the new format.

The ACLI verify-config command warns you if the system has loaded a configuration containing illegal naming syntax.

Please note that the software allows you to make changes to HMRs, including configuring new functionality to existing rules, as long as you do not change the rule name. This results in an important consideration surrounding HMRs with hyphens in previously configured rule names.

- You can reference stored values in new value names. (Recall that stored values may be rule names.)
- You can perform subtraction in new value names.

If you use a rule names with hyphens within the REGEX of new value names, the system cannot determine whether the hyphen is part of the rule name or is intended to invoke subtraction within the REGEX. For this reason, you need to use great care with legacy HMR naming that includes hyphens.

As a general rule, create new rule names that follow the new rule naming guidelines if you intend to use new functionality in those rules.

New Value Restrictions

To simplify configuration and remove possible ambiguity, the use of boolean and equality operators (==, <=, <, etc.) for new-value parameter values has been banned. Since there was no specific functionality tied to their use, their ceasing to be use will have no impact to normal SIP manipulation operations.

Header Manipulation Rules for SDP

The Oracle Enterprise Session Border Controller supports SIP header and parameter manipulation rules for four types of SIP message contents:

- headers
- elements within headers
- ASCII-encoded Multipurpose Internet Mail Extensions (MIME) bodies
- binary-encoded MIME ISDN User Part (ISUP) bodies

While Session Description Protocol (SDP) offers and answers can be manipulated in a fashion similar to ASCII-encoded MIME, such manipulation is primitive in that it lacks the ability to operate at the SDP session- and media-levels.

In addition, the system supports a variant of Header Manipulation Rules (HMR) operating on ASCII-encoded SDP bodies, with specific element types for descriptors at both the session-level and media-level, and the ability to apply similar logic to SDP message parts as is done for SIP header elements.

The configuration object, mime-sdp-rules, under sip-manipulation specifically addresses the manipulation of SDP parts in SIP messages. Just as existing header-rules are used to manipulate specific headers of a SIP message, mime-sdp-rules will be used to manipulate the SDP specific mime-attachment of a SIP message.

Platform Support

HMR for SDP is available on the following platforms: Acme Packet 38x0 and Acme Packet 4500.

SDP Manipulation

mime-sdp-rules function in a similar fashion as header-rules. They provide

- parameters used to match against specific SIP methods and/or message types
- parameters used to match and manipulate all or specified parts of an SDP offer or answer
- a means of comparing search strings or expressions against the entire SDP
- different action types to allow varying forms of manipulation

Since only a single SDP can exist within a SIP message, users need not specify a content-type parameter as is necessary for a mime-rule. A mime-sdp-rule operates on the single SDP within the SIP message. If no SDP exists with the message, one can be added. If the message already contains a mime attachment, adding SDP results in a multipart message.

All manipulations performed against all or parts of the SDP are treated as UTF-8 ASCII encoded text. At the parent-level (mime-sdp-rule) the match-value and new-value parameters execute against the entire SDP as a single string.

An add action only succeeds in the absence of SDP because a message is allowed only a single SDP offer or answer. A delete operation at the mime-sdp-rule level will remove the SDP entirely.

Note that on an inbound sip-manipulation, SDP manipulations interact with the Oracle Enterprise Session Border Controller codec-policy. SDP manipulations also interact with codec reordering and media setup. It is very possible to make changes to the SDP such that the call can not be setup due to invalid media parameters, or settings that will affect the ability to transcode the call. Consequently, user manipulation of the SDP can prove risky, and should be approached with appropriate caution.

Three configuration-objects, sdp-session-rule, sdp-media-rule, and mime-header-rule, exist under the mime-sdp-rule. These objects provide finer grained control of manipulating parts of the SDP.

sdp-session-rule

An sdp-session-rule groups all SDP descriptors, up until the first media line, into a single entity, thus allowing the user to perform manipulation operations on a session-specific portion of the SDP.

Like the mime-sdp-rule, all match-value and new-value operations performed at this level are executed against the entire session group as a complete string. Given the sample SDP below, if an sdp-session-rule is configured, the match-value and new-values operate only on the designated portion.

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
```

```
        m=audio 49170 RTP/AVP 0
        m=video 51372 RTP/AVP 31
        m=application 32416 udp wb
        a=orient:portrait
```

Nested under the sdp-session-rule configuration object is an sdp-line-rule object, the object that identifies individual descriptors within the SDP. The types of descriptors used at the sdp-session-rule level are v, o, s, i, u, e, p, c, b, t, r, z, k, and a, the descriptors specific to the entire session description.

This level of granularity affords the user a very simple way to making subtle changes to the session portion of the SDP. For instance, it is very common to have to change the connection line at the session level.

The add and delete actions perform no operation at the sdp-session-rule level.

sdp-media-rule

An sdp-media-rule groups all of the descriptors that are associated with a specific media-type into single entity, thus allowing the user to perform manipulation operations on a media-specific portion of the SDP. For example, a user can construct an sdp-media-rule to change an attribute of the audio media type.

Like a mime-sdp-rule, all match-value and new-value operations performed at this level are executed against the entire media-group as a complete string. Given the sample SDP below, if a media-level-descriptor is configured to operate against the application group, the match-value and new-values would operate only on designated portion.

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        s=SDP Seminar
        i=A Seminar on the session description protocol
        u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
        e=mjh@isi.edu (Mark Handley)
        c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
        a=recvonly
        m=audio 49170 RTP/AVP 0
        m=video 51372 RTP/AVP 31
        m=application 32416 udp wb
        a=orient:portrait
```

A configuration parameter media-type is used to specify the media group on which to operate. It contains all of the descriptors including the m-line up to the next m-line. This parameter is a string field and must match the media-type exactly as it appears within the SDP. The special media-type media can be used to refer to all media types. This is particularly useful when performing an add operation, when the user wants to add a media section between the first and second medias, but does not know what media type they are. Otherwise, during an add operation, the media section would be added before the specified media-type (if no index parameter was provided).

The types of descriptors used at the sdp-media-rule level are m, i, c, b, k, and a, the descriptors specific to the media description.

This level of granularity affords the user a very simple way to making subtle changes to the media portion of the SDP. For instance, it is very common to have to change the name of an audio format (for example G729 converted to g729b), or to add attributes specific to a certain media-type.

The index operator is supported for the media-type parameter (for example, media-type audio[1]). Like header rules, if no index is supplied, this means operate on all media-types that match the given name. For specifying specific media-types, the non-discrete indices are also supported (for example, ^ - last). Adding a media-type, without any index supplied indicates that the media should be added at the beginning. The special media-type media uses the index as an absolute index to all media sections, while a specific media-type will index relative to that given media type.

For sdp-media-rules set to an action of add where the media-type is set to media, the actual media type is obtained from the new-value, or more specifically, the string after m= and before the first space.

Given the following SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
        m=audio 48324 RTP/AVP 8
        m=video 51372 RTP/AVP 31
```

With the sdp-media-rule:

```
sdp-media-rule
        name                        smr
        media-type                  audio[1]
        action                      manipulate
        comparison-type             case-sensitive
        match-value
        new-value                   "m=audio 1234 RTP/AVP 8 16"
```

This rule operates on the 2nd audio line, changing the port and adding another codec, resulting in the SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
        m=audio 1234 RTP/AVP 8 16
        m=video 51372 RTP/AVP 31
```

The following rule, however:

```
sdp-media-rule
        name                        smr
        media-type                  media[1]
        action                      add
        comparison-type             case-sensitive
        match-value
        new-value                   "m=video 1234 RTP/AVP 45"
```

adds a new video media-type at the 2nd position of all media-lines, resulting in the SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
        m=video 1234 RTP/AVP 45
        m=audio 48324 RTP/AVP 8
        m=video 51372 RTP/AVP 31
```

sdp-line-rule

Unlike header-rules, sdp descriptors are not added in the order in which they are configured. Instead they are added to the SDP adhering to the grammar defined by RFC 4566 (as is shown below).

```
      Session description
         v=  (protocol version)
         o=  (originator and session identifier)
         s=  (session name)
         i=* (session information)
         u=* (URI of description)
         e=* (email address)
         p=* (phone number)
            c=* (connection information -- not required if included in
              all media)
         b=* (zero or more bandwidth information lines)
         One or more time descriptions ("t=" and "r=" lines; see
              below)
```

```
         z=* (time zone adjustments)
         k=* (encryption key)
         a=* (zero or more session attribute lines)
         Zero or more media descriptions (see below)

      Time description
         t=  (time the session is active)
         r=* (zero or more repeat times)

      Media description, if present
         m=  (media name and transport address)
         i=* (media title)
         c=* (connection information -- optional if included at
              session level)
         b=* (zero or more bandwidth information lines)
         k=* (encryption key)
         a=* (zero or more media attribute lines)
```

* after the equal sign denotes an optional descriptor.

This hierarchy is enforced meaning that if you configure a rule which adds a session name descriptor followed by a rule which adds a version descriptor, the SDP will be created with the version descriptor first, followed by the session name.

The only validation that will occur is the prevention of adding duplicate values. In much the same way that header-rules prevents the user from adding multiple To headers, the descriptor rule will not allow the user to add multiple descriptors; unless multiple descriptors are allowed, as is in the case of b, t, r and a.

There exists a parameter type under the sdp-line-rule object that allows the user to specify the specific line on which to perform the operation. For example: v, o, s, i, u, e, p, c, b, t, r, z, k, a, and m. Details on these types can be found in RFC 4566.

For those descriptors, of which there may exist zero or more (b, t, r, and a) entries, indexing grammar may be used to reference the specific instance of that attribute. This indexing grammar is consistent with that of header-rules for referring to multiple headers of the same type.

Given the example SDP below:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        s=SDP Seminar
        i=A Seminar on the session description protocol
        u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
        e=mjh@isi.edu (Mark Handley)
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        r=604800 3600 0 90000
        r=7d 1h 0 25h
        a=recvonly
        m=audio 49170 RTP/AVP 0
        m=video 51372 RTP/AVP 31
        m=application 32416 udp wb
        a=orient:portrait
```

and the following sdp-line-rule:

```
        sdp-line-rule
           name                 removeRepeatInterval
           type                 r[1]
           action               delete
```

The rule removeRepeatInterval removes the second repeat interval descriptor within the SDP.

The behavior of all SDP rules follow the same behavior of all manipulation rules in that they are executed in the order in which they are configured and that each rule executes on the resultant of the previous rule.

Each descriptor follows its own grammar and rules depending on the type specified. The values of the descriptor are evaluated at runtime since the new-values themselves are evaluated at runtime. At this time no validation of the grammar for each of the types is performed. The user is responsible for properly formatting each of the descriptors according to their specifications.

For instance, the version (v) descriptor can be removed from the SDP but leaving all descriptors for that SDP, causing the SDP to become invalid. This is consistent with the way header-rules operate, in that there is no validation for the specific headers once they have been manipulated through HMR.

Regular Expression Interpolation

An interpolated regular expression is a regular expression that is compiled and evaluated at runtime. Today all regular expressions are compiled at configuration time in order to improve performance. There are cases where a regular expression is determined dynamically from data within a SIP message. In these circumstances the regular expression is unknown until the time of execution.

In order to have a regular expression be interpolated at runtime, it must be contained within a set of {}. An interpolated expression can have any number of regular expressions and strings appended together. Any characters to the left or right of the curly braces will be appended to the value within the curly braces. The curly braces are effectively two operators treated as one (interpolate the value contained within and then concatenate the values to the left and right of the curly braces). If the comparison-type is set to pattern-rule and the match-value contains a value that matches the grammar below, then it will be treated as an interpolated expression.

([^\\]|^)\{\$[^0-9]+[^}]*\}

The example below demonstrates using a user defined variable within a regular expression of another rule at runtime.

```
element-rule

        name                            someRule
        type                            header-value
        action                          replace
        comparison-type                 pattern-rule
        match-value                     ^sip:{$rule1.$0}@(.+)$
        new-value                       sip:bob@company.com
```

If the value of $rule1.$0 evaluates to alice then it will successfully match against the string sip:alice@comcast.net. An interpolated expression can be as simple as “{$rule1.$0}” or as complex as ^sip:{rule1.$0}@{$rule2[1].$2}$. It is not possible to interpolate a normal regular expression since the grammar will not allow the user to enter such an expression. Only variables can be contained with the curly braces.

The resultant of interpolated expressions can be stored in user defined variables. Given the same example from above, if the rule someRule was referenced by another rule, the value of sip:alice@comcast.net would be stored within that rule.

Interpolation only makes a single pass at interpolation, but does so every time the Rule executes. In other words, if the Rule is applied to the Route header, it will interpolate again for each Route header instance. What this means is that the value within the curly braces will only be evaluated once. For instance, if the value {$someRule.$1} evaluates to {$foobar.$2} the Oracle Enterprise Session Border Controller (E-SBC) will treat $foobar.$2 as a literal string which it will compile as a regular expression. The E-SBC will not recursively attempt to evaluate $foobar.$2, even if it was a valid user defined variable.

Interpolated regular expressions will evaluate to TRUE if an only if both the regular expression itself can be compiled and it successfully matches against the compared string.

Regular Expressions as Boolean Expressions

Regular expressions can be used as boolean expressions today if they are the only value being compared against a string, as is shown in the case below.

```
mime-rule
        name                            someMimeRule
        content-type                    application/text
```

```
        action                          replace
        comparison-type                 pattern-rule
        match-value                     ^every good boy .*
        new-value                       every good girl does fine
However, regular expressions can not be used in conjunction with other
boolean expressions to form more complex boolean expressions, as is shown
below.

mime-rule
     name                       someMimeRule
     content-type               application/text
     action                     replace
     comparison-type            boolean
     match-value                $someRule & ^every good boy .*
     new-value                  every good girl does fine
```

There are many cases where the user has the need to compare some value as a regular expression in conjunction with another stored value. It is possible to perform this behavior today, however it requires an extra step in first storing the value with the regular expression, followed by another Manipulation Rule which compares the two boolean expressions together (e.g. $someRule & $someMimeRule).

In order to simplify the configuration of some sip-manipulations and to make them more efficient this functionality is being added.

Unfortunately, it is not possible to just use the example as is shown above. The problem is there are many characters that are commonly used in regular expressions that would confuse the HMR expression parser (such as $, and +). Therefore delimiting characters need to be used to separate the regular expression from the other parts of the expression.

To treat a regular expression as a boolean expression, it needs to be enclosed within the value $REGEX(<expression>,<compare_string>=$ORIGINAL); where <expression> is the regular expression to be evaluated. <compare_string> is the string to compare against the regular expression. This second argument to the function is defaulted to $ORIGINAL which is the value of the of the specific Manipulation Rule object. It can be overridden to be any other value the user desires.

The proper configuration for the example above to use regular expressions as boolean expressions is

```
mime-rule
        name              someMimeRule
        content-type      application/text
        action            replace
        comparison-type   boolean
        match-value       $someRule & $REGEX("^every good boy .*")
        new-value         every good girl does fine
```

It is also possible to use expressions as arguments to the $REGEX function. These expressions will in turn be evaluated prior to executing the $REGEX function. A more complex example is illustrated below.

```
header-rule
        name               checkPAU
        header-name        request-uri
        action             reject
        comparison-type    boolean
        match-value        (!$REGEX($rule1[0],$FROM_USER))&
                           (!$REGEX($rule2[0],$PAI_USER))
        msg-type           request
        new-value          403:Forbidden
        methods            INVITE,SUBSCRIBE,MESSAGE,PUBLISH,
                           OPTIONS, REFER
```

It should be noted that when using $REGEX() in a boolean expression, the result of that expression is not stored in the user variable. The comparison-type must be set to pattern-rule in order to store the result of a regular expression.

The arguments to the $REGEX() function are interpolated by default. This is the case since the arguments themselves must be evaluated at runtime. The following example is also valid.

```
mime-rule
        name                    someMimeRule
        content-type            application/text
        action                  replace
        comparison-type         boolean
        match-value             $someRule & $REGEX("^every good
                                {$rule1[0].$0} .*")
```

Moving Manipulation Rules

Users can move rules within any manipulation-rule container. Any manipulation rule which contains sub-rules will now offer the ACLI command move <from index> <to index>. For example, given the order and list of rules below:

1. rule1

2. rule2

3. rule3

4. rule4

Moving rule3 to position 1 can be achieved by executing move 3 1. The resulting order will then be: rule3, rule1, rule2, rule4. A move operation causes a shift (or insert before) for all other rules. If a rule from the top or middle moves to the bottom, all rules above the bottom are shifted up to the position of the rule that was moved. If a rule from the bottom or middle moves to the top, all rules below are shifted down up to the position of the rule that was moved. Positions start from 1.

A valid from-index and to-index are required to be supplied as arguments to the move action. If a user enters a range that is out of bounds for either the from-index or to-index, the ACLI will inform the user that the command failed to execute and for what reason.

With respect to the issue of creating an invalid sip-manipulation by incorrectly ordering the manipulation rules, this issue is handled by the Oracle Enterprise Session Border Controller validating the rules at configuration time and treating them as invalid prior to runtime. This may or may not affect the outcome of the sip-manipulation as a configured rule may not perform any operation if it refers to a rule that has yet to be executed. It is now the user's responsibility to reorder the remaining rules in order to make the sip-manipulation valid once again.

It is important to note that rules of a different type at the same level are all part of the same list. To clarify; header-rules, mime-rules, mime-isup-rules and mime-sdp-rules all share the same configuration level under sip-manipulation. When selecting a move from-index and to-index for a header-rule, one must take into consideration the location of all other rules at the same level, since the move is relative to all rules at that level, and not relative to the particular rule you have selected (for example, the header-rule).

Since the list of rules at any one level can be lengthy, the move command can be issued one argument at a time, providing the user with the ability to select indices. For instance, typing move without any arguments will present the user with the list of all the rules at that level. After selecting an appropriate index, the user is then prompted with a to-index location based on the same list provided.

For Example:

```
ACMEPACKET(sip-mime-sdp-rules)# move
select a rule to move
```

1: msr sdp-type=any; action=none; match-value=; msg-type=any

2: addFoo header-name=Foo; action=none; match-value=; msg-type=any

3: addBar header-name=Bar; action=none; match-value=; msg-type=any

```
selection: 2
destination: 1
Rule moved from position 2 to position 1
ACMEPACKET(sip-mime-sdp-rules)#
```

Rule Nesting and Management

There will be cases where the user wants to take a stored value from the SDP and place it in a SIP header, and vice-versa. All header-rules, element-rules, mime-rules, mime-isup-rules, isup-param-rules, mime-header-rules and mime-sdp-rules are inherited from a Manipulation Rule. A Sip Manipulation is of type Manipulation which contains a list of Manipulation Rules. Each Manipulation Rule can itself contain a list of Manipulation Rules. Therefore when configuring manipulation rules, they will be saved in the order which they have been configured. This is different from the way other configuration objects are configured. Essentially, the user has the option of configuring which type of object they want and when they are done, it gets added to the end of the sip-manipulation, such that order is preserved. This will mean that any Manipulation Rule at the same level can not share the same name. For example, names of header-rules can't be the same as any of the mime-sdp-rule ones or mime-isup-rule. This allows the user to reference stored values from one rule type in another at the same level.

ACLI Configuration Examples

The following eight sections provide sample SDP manipulations.

Remove SDP

```
sip-manipulation
        name                        stripSdp
        description                 remove SDP from SIP message
        mime-sdp-rule
                name                    sdpStrip
                msg-type                request
                methods                 INVITE
                action                  delete
                comparison-type         case-sensitive
                match-value
                new-value
```

Remove Video from SDP

```
sip-manipulation
        name                          stripVideo
        description                   strip video codecs from SIP
                                      message
        mime-sdp-rule
                name                      stripVideo
                msg-type                  request
                methods                   INVITE
                action                    manipulate
                comparison-type           case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                 removeVideo
                        media-type           video
                        action               delete
                        comparison-type      case-sensitive
match-value
                        new-value
```

Add SDP

```
sip-manipulation
        name                        addSdp
        description                 add an entire SDP if one does
                                    not exist
        mime-sdp-rule
                name                    addSdp
                msg-type                request
                methods                 INVITE
```

```
                action                  add
                comparison-type         case-sensitive
                match-value
                new-value               "v=0\r\no=mhandley
2890844526 2890842807 IN IP4 "+$LOCAL_IP+"\r\ns=SDP Seminar\r\ni=A
Seminar on the session description protocol\r\nu=http:
//www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps\r\ne=mjh@isi.edu
(Mark Handley)\r\nc=IN IP4 "+$LOCAL_IP+"\r\nt=2873397496
2873404696\r\na=recvonly\r\nm=audio 49170 RTP/AVP 0\r\n"
```

Manipulate Contacts

This rule changes the contact in the SDP to the value contained in the Contact header.

```
sip-manipulation
        name                            changeSdpContact
        description                     changes the contact in the SDP to the
value of the contact header
        header-rule
                name                            storeContact
                header-name                     Contact
                action                          store
                comparison-type                 pattern-rule
                msg-type                        request
                methods                         INVITE
                match-value
                new-value
                element-rule
                        name                            storeHost
                        parameter-name
                        type                            uri-host
                        action                          store
                        match-val-type                  ip
                        comparison-type                 pattern-rule
                        match-value
                        new-value
        mime-sdp-rule
                name                            changeConnection
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-session-rule
                        name                    changeCLine
                        action                  manipulate
                        comparison-type         case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                           name                 updateConnection
                           type                 c
                           action               replace
                           comparison-type      case-sensitive
                           match-value          $storeContact.$storeHost
                           new-value            $storeContact.$storeHost.$0
```

Remove a Codec

This rule changes the contact in the SDP to the value contained in the Contact header.

```
sip-manipulation
        name                            removeCodec
        description                     remove G711 codec if it exists
```

```
        mime-sdp-rule
                name                            removeCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                        removeG711
                        media-type                  audio
                        action                      manipulate
                        comparison-type             case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name                    remove711
                                type                    m
                                action                  replace
                                comparison-type         pattern-rule
                                match-value             ^(audio [0-9]
                                                        {1,5} RTP.*)( [07]
                                                        \b)(.*)$
                                new-value               $1+$3
                        sdp-line-rule
                                name                    stripAttr
                                type                    a
                                action                  delete
                                comparison-type         pattern-rule
                                match-value             ^(rtpmap|fmtp):
                                                        [07]\b$
                                new-value
```

Change Codec

```
sip-manipulation
        name                            convertCodec
        description                     changeG711toG729
        mime-sdp-rule
                name                            changeCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                      change711to729
                        media-type                audio
                        action                    manipulate
                        comparison-type           case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name                  change711
                                type                  m
                                action                replace
                                comparison-type       pattern-rule
                                match-value           ^(audio [0-9]{4,5}
                                                      RTP/AVP.*)( 0)(.*)$
new-value            $1+” 18”+$3
                        sdp-line-rule
                                name                  stripAttr
                                type                  a
```

```
                                action                delete
                                comparison-type       pattern-rule
                                match-value           ^rtpmap:0 PCMU/
                                                      .+$
                                new-value
                        sdp-line-rule
                                name                  addAttr
                                type                  a
                                action                add
                                comparison-type       boolean
                                match-value           $change711to729.
                                                      $stripAttr
                                new-value             rtpmap:18 G729/8000
```

Remove Last Codec and Change Port

```
sip-manipulation
        name                            removeLastCodec
        description                     remove the last codec
        mime-sdp-rule
                name                            removeLastCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                       removeLast
                        media-type                 audio
                        action                     manipulate
                        comparison-type            case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name                    isLastCodec
                                type                    m
                                action                  store
                                comparison-type         pattern-rule
                                match-value             ^(audio )([0-9]{4,
                                                        5})( RTP/AVP
                                                        [0-9]{1-3})$
new-value
                        sdp-line-rule
                                name                    changePort
                                type                    m
                                action                  replace
                                comparison-type         boolean
                                match-value             $removeLastCodec.
$removeLast.$isLastCodec
                                new-value               $removeLastCodec.
$removeLast.$isLastCodec.$1+0+$removeLastCodec.$removeLast.
$isLastCodec.$3
```

Remove Codec with Dynamic Payload

```
sip-manipulation
        name                            removeAMR
        description                     remove the AMR and AMR-WB dynamic
codecs
        mime-sdp-rule
                name                            sdpAMR
                msg-type                        request
                methods                         INVITE
```

```
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
name                      mediaAMR
                        media-type                audio
                        action                    manipulate
                        comparison-type           case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name              isAMR
                                type              a
                                action            delete
                                comparison-type   pattern-rule
                                match-value       ^rtpmap:([0-9]
                                                  {2,3}) AMR
                                new-value
                sdp-media-rule
                        name                      mediaIsAMR
                        media-type                audio
                        action                    manipulate
                        comparison-type           boolean
                        match-value               $sdpAMR.$media
                                                  AMR.$isAMR
                        new-value
                        sdp-line-rule
                                name              delFmtpAMR
                                type              a
                                action            delete
                                comparison-type   pattern-rule
                                match-value       ^fmtp:{$sdpAMR.
                                                  $mediaAMR.
                                                  $isAMR.$1}\b
                                new-value
                        sdp-line-rule
                               name             delAMRcodec
                               type             m
                               action           find-replace-all
                                comparison-type pattern-rule
                                match-value     ^audio [0-9]+
                                                RTP.*( {$sdpAMR.
                                                 $mediaAMR.$isAMR.
```

Dialog Transparency

This section explains how to configure dialog transparency, which prevents the Oracle Enterprise Session Border Controller from generating a unique Call-ID and modifying dialog tags.

Overview

With dialog transparency enabled, the Oracle Enterprise Session Border Controller is prevented from generating a unique Call-ID and from modifying the dialog tags; the Oracle Enterprise Session Border Controller passes what it receives. Therefore, when a call made on one Oracle Enterprise Session Border Controller is transferred to another UA and crosses a second Oracle Enterprise Session Border Controller, the second Oracle Enterprise Session Border Controller does not note the context of the original dialog, and the original call identifiers are preserved end to end. The signalling presented to each endpoint remains in the appropriate context regardless of how many times a call crosses through a Oracle Enterprise Session Border Controller or how many Oracle Enterprise Session Border Controllers a call crosses.

Without dialog transparency enabled, the Oracle Enterprise Session Border Controller's SIP B2BUA rewrites the Call-ID header and inserted dialog cookies into the From and To tags of all messages it processes. These dialog cookies are in the following format: SDxxxxxNN-. Using these cookies, the Oracle Enterprise Session Border Controller can recognize the direction of a dialog. However, this behavior makes call transfers problematic because one Oracle Enterprise Session Border Controllers' Call-ID might not be properly decoded by another Oracle Enterprise Session Border Controller. The result is asymmetric header manipulation and failed call transfers.

Dialog Transparency Configuration

You set one parameter in your SIP configuration to enable dialog transparency.

- For new configurations, this feature defaults to enabled

To enable SIP dialog transparency:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
```

4. Use the ACLI select command so that you can work with the SIP configuration.

```
ACMEPACKET(sip-config)# select
```

5. dialog-transparency—Enter the state of SIP dialog transparency you require for your Oracle Enterprise Session Border Controller. The default value is enabled. The valid values are:
   - enabled | disabled

Route Header Removal

This section explains how to enable the Oracle Enterprise Session Border Controller to disregard and strip all SIP Route headers. You set an option in a SIP interface configuration to strip all Route headers for SIP requests coming from this interface.

When the Oracle Enterprise Session Border Controller with this option configured receives an INVITE from an interface, it removes the route headers. However, although it removes the headers, the Oracle Enterprise Session Border Controller maintains backward compatibility with RFC 2543 nodes. To do so, it normalizes the request to an RFC 3261 loose routing form before it removes the headers.

Route Header Removal Configuration

The following information explains how to remove SIP route headers.

To configure SIP route header removal:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Type options strip-route-headers and press Enter. This completes the configuration of SIP route header removal.

```
ACMEPACKET(sip-interface)# options strip-route-headers
```

SIP Via Transparency

This section explains the inbound Via header transparency feature, which enables the Oracle Enterprise Session Border Controller to insert its Via header on top of the top-most Via header received from user equipment (UE). It then forwards it on to the IP Multimedia Subsystem (IMS) core with the original Via header now located as the bottom-most Via header.

The Oracle Enterprise Session Border Controller still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.

This feature is targeted for the Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) with SIP hosted NAT traversal support. It works with SIP NAT bridged, local-policy routed, and non-SIP NAT configurations, regardless of registration handling.

Some equipment acts as Proxy-CSCF (P-CSCF) and Serving-CSCF (S-CSCF) nodes, with the Oracle Enterprise Session Border Controller is located between the equipment and user endpoints. The equipment needs to see the each user endpoint's original Via header in order to perform some implicit authentication, admission, and control functions in a TISPAN-compliant model.

You enable Via header transparency on the access SIP interface. Received Via headers are saved for inclusion in requests going out another interface or session agent that does not have the parameter set, in other words, the core side. For any received SIP message where the inbound previous hop interface was enabled for Via header transparency, theOracle Enterprise Session Border Controller adds its own Via header as it forwards it, and it also copies the received top-most Via as the new bottom-most Via, if the outbound next hop interface/session agent is not enabled for Via header transparency. The Oracle Enterprise Session Border Controller also adds a received= parameter to the copied Via header, per the SIP RFC 3261.

Any message received from an interface without Via header transparency enabled, does not have the received Via header copied over to any other direction.

For HNT, where the original top-most (and only) Via header from a UE is a private/false address, the SD should still copy that false address into the core-side, and the received= parameter will contain the real Layer-3 addressing.

SIP Via Transparency Configuration

You can configure SIP Via header transparency for the access SIP interface using the ACLI.

To configure SIP Via header transparency for an access interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. You can either add support to a new SIP interface configuration or to an existing SIP interface configuration:

   For a new SIP interface configuration, you can add the option by typing options, a Space, and then via-header-transparency.

```
ACMEPACKET(sip-interface)# options via-header-transparency
```

For an existing SIP interface configuration without options configured, select the SIP interface, type options followed by a Space, and then via-header-transparency.

```
ACMEPACKET(sip-interface)# select
ACMEPACKET(sip-interface)# options via-header-transparency
```

For an existing SIP interface configuration with options configured, select the SIP interface, type options followed by a Space, the plus sign (+), and the via-header-transparency option.

```
ACMEPACKET(sip-interface)# select
ACMEPACKET(sip-interface)# options +via-header-transparency
```

5. Save your work using the ACLI save or done command.

Symmetric Latching

Symmetric latching, or forced HNT, ensures that symmetric RTP/RTCP is used for a SIP endpoint. Symmetric RTP/RTCP means that the IP address and port pair used by an outbound RTP/RTCP flow is reused for the inbound flow. The IP address and port are learned when the initial RTP/RTCP flow is received by the Oracle Enterprise Session Border Controller. The flow's source address and port are latched onto and used as the destination for the RTP/RTCP sourced by the other side of the call. The IP address and port in the c line and m line respectively in the SDP message are ignored.

If your network is configured with nested realms in order to separate signalling from media, make sure that the symmetric latching feature is enabled on the signaling realm.

Note: This description is applicable to RTCP only when you also enable the HNT RTCP option in the media-manager configuration. Do not enable symmetric latching on core-facing interfaces.

Symmetric Latching Configuration

To configure symmetric latching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>           0.0.0.0
2: MGCP_Realm <none>           0.0.0.0
3: H323REALM  <none>           0.0.0.0
selection: 1
ACMEPACKET(realm-config)#
```

5. symmetric-latching — Enable symmetric latching on the SBC. This completes the configuration of forced HNT. The default value for this parameter is disabled. The valid values are:

   - enabled | disabled

   ```
   ACMEPACKET(realm-config)# symmetric-latching enabled
   ```

6. Save your work using the ACLI save or done command.

Enabling RTCP Latching

To enable RTCP symmetric latching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type media-manager and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. Select the media manager configuration so that you can enable HNT RTCP.

```
ACMEPACKET(media-manager-config)# select
```

5. hnt-rtcp — Enable support of RTCP when the SBC performs HNT. The default value is disabled. The valid values are:

   - enabled | disabled

```
ACMEPACKET(media-manager-config)# hnt-rtcp enabled
```

6. Save your work using either the ACLI save or done command.

SIP Number Normalization

This section explains the SIP number normalization feature that applies to the SIP To URI. (Currently the Oracle Enterprise Session Border Controller supports number normalization on From and To addresses for both inbound and outbound call legs.) Number normalization includes add, delete, and replace string functions that result in consistent number formats.

Number normalization is supported for the following call types:

- SIP to SIP
- H.323 to SIP

Number normalization applies to the SIP To URI. It occurs on ingress traffic, prior to the generation of accounting records or local policy lookups. RADIUS CDR attributes are populated with the normalized numbers. Local policy matching is based on the normalized numbers.

Terminology

The following lists explains the terminology used later.

- X is any digit having the value 0 through 9
- N is any digit having the value 2 through 9
- 0/1 is a digit having the value of either 0 or 1
- NXX is a form of Numbering Plan Area (NPA).
- CC is a 1, 2, or 3 digit country code used in international dialing
- NN is a national number that can be a four to fourteen digit national number used in international dialing, where the combination of CC+NN is a 7 to 15 digit number.
- + symbol in E.164 indicates that an international prefix is required
- E.164 numbers are globally unique, language independent identifiers for resources on Public Telecommunication Networks that can support many different services and protocols.
- N11 number is any of the three-digit dialing codes in the form N11 used to connect users to special services, where N is a digit between 2 and 9

Calls from IP Endpoints

The Oracle Enterprise Session Border Controller uses the following number normalization rules:

- North American Numbering Plan (NANP) calls: where a number with the format 1NPANXXXXXX is received, the Oracle Enterprise Session Border Controller adds a plus sign (+) as a prefix to the NANP number. The Oracle Enterprise Session Border Controller also adds the string ;user=phone after the host IP address in the SIP URI. For example:

  ```
  sip:+1NPANXXXXXX@ipaddr;user=phone
  ```
- International NWZ1 calls: Oracle Enterprise Session Border Controller receives an international call with the format 011CCNN. The Oracle Enterprise Session Border Controller deletes the 011 prefix and adds a plus sign (+) as a prefix to CC+NN; and also adds the string ;user=phone after the host IP address in the SIP URI. For example:

  ```
  sip:+CCNN@ipaddr;user=phone
  ```
- Private number calls: when a private number with the format nxxxx (where n=2 through 9) is received, no number normalization is applied by the Oracle Enterprise Session Border Controller.
- Calls to numbers such as N11, 0-, 0+, 00-, and 01+: the Oracle Enterprise Session Border Controller adds ;phone-context=+1 after the number and also adds the string ;user=phone after the host IP address in the SIP URI. For example:

  ```
  sip:N11;phone-context=+1@ipaddr;user=phone
  sip:01CCNN;phone-context=+1@ipaddr;user=phone
  ```
- Calls with numbers that are already normalized are not modified by the Oracle Enterprise Session Border Controller.

Calls from IP Peer Network

For calls received from external peer networks, the Oracle Enterprise Session Border Controller uses the following number normalization rules:

- Global numbers such as NANP and international E.164 numbers should have already been normalized. If not, the Oracle Enterprise Session Border Controller applies the same number normalization rules listed in the prior section.
- Calls to numbers such as N11, 0-, 0+, 00-, and 01+: the Oracle Enterprise Session Border Controller adds ;phone-context=+1 after the number and also adds the string ;user=phone (if absent) after the host IP address in the SIP URI.

SIP Number Normalization Configuration

You can configure SIP number normalization for the realm and session agent using the ACLI.

Realm

To configure SIP number normalization for a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type media-manager and press Enter to access the media-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```
3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```
4. You can either add SIP number normalization support to a new session agent configuration or to an existing session agent configuration:

   - For a new realm configuration, add the option by typing options, a Space, and then number-normalization.

```
ACMEPACKET(realm-config)# options number-normalization
```

- For an existing realm configuration without any options already configured, select the realm, type options followed by a Space, and then number-normalization.

```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)# options number-normalization
```

- For an existing realm configuration with other options, select the realm, type options followed by a Space, the plus sign (+), and the number-normalization option.

```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)# options +number-normalization
```

5. Save your work using the ACLI save or done command.

Session Agent

To configure SIP number normalization for a session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. You can either add SIP number normalization support to a new session agent configuration or to an existing session agent configuration:

   - For a new a session agent configuration, add the option by typing options, a Space, and then number-normalization.

```
ACMEPACKET(session-agent)# options number-normalization
```

   - For an existing session agent configuration without any options already configured, select the session agent, type options followed by a Space, and then number-normalization.

```
ACMEPACKET(session-agent)# select
ACMEPACKET(session-agent)# options number-normalization
```

   - For an existing session agent configuration with other options, select the session agent, type options followed by a Space, the plus sign (+), and the number-normalization option.

```
ACMEPACKET(session-agent)# select
ACMEPACKET(session-agent)# options +number-normalization
```

5. Save your work using the ACLI save or done command.

SIP Port Mapping

This section contains information about the SIP port mapping feature. SIP port mapping lets you allocate a unique SIP signaling transport address (IP address and UDP port) on the Oracle Enterprise Session Border Controller in the provider network for each registered endpoint (user agent).

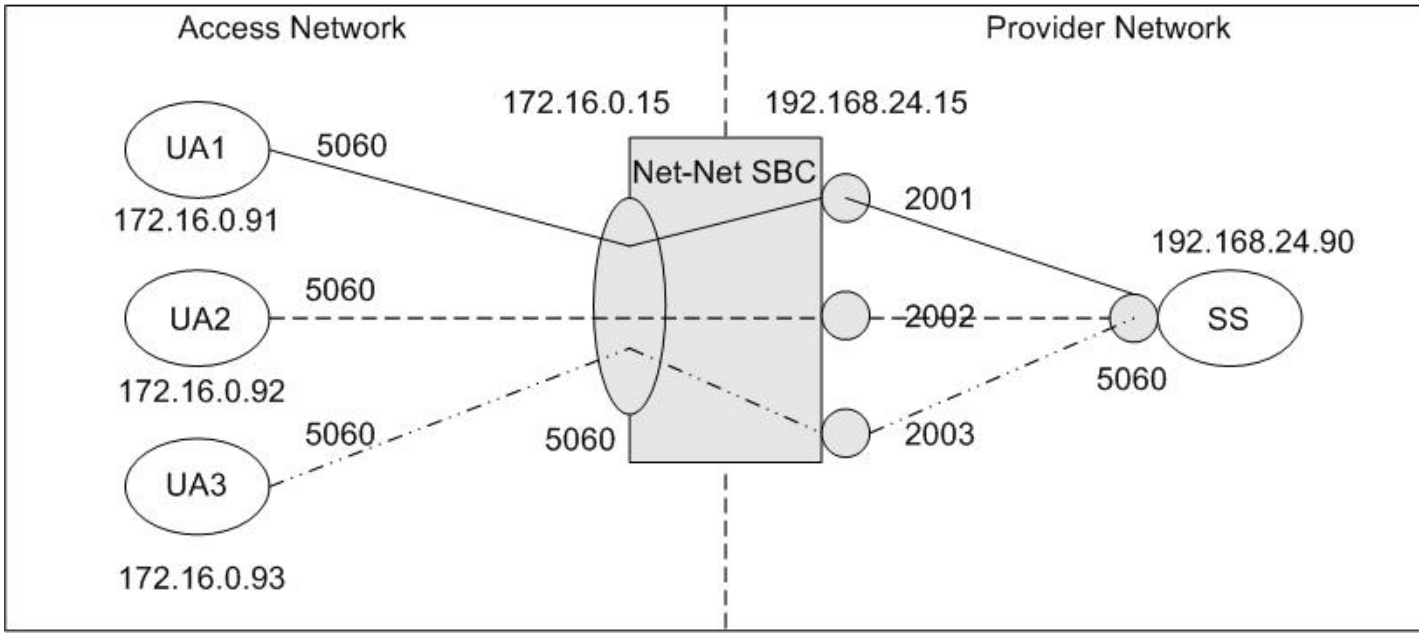
About SIP Port Mapping

You might need to provide a unique signaling transport address for each registered endpoint for admission control, if required by your softswitch vendor. If you have questions about your softswitch, contact the vendor for assistance.

When a Oracle Enterprise Session Border Controller resides between the endpoints and the softswitch, the softswitch sees the same transport address (that of the Oracle Enterprise Session Border Controller) for all endpoints. By

allocating a unique UDP port for each endpoint, the Oracle Enterprise Session Border Controller provides each of them a unique transport address.

The following example illustrates the SIP port mapping feature.



The diagram shows UA1, UA2, and UA3 are endpoints within the access network and that the SIP interface for the access network is 172.16.0.15:5060. On the provider network, the SIP interface is at 192.168.24.15, with the SIP port mapping feature enabled. The softswitch/registrar is also located on the provider network at 192.168.24.90:5060.

The diagram shows that port 2001 on the provider network is allocated to UA1 on the access network, port 2002 is allocated to UA2, and port 2003 is allocated to UA3. Because of this allocation, all SIP signaling messages sent from the endpoints in the access network to the softswitch on the provider network travel through an allocated signaling port. For example, all signaling messages between UA1 and the softswitch use 192.168.24.15:2001 as the transport address.

How SIP Port Mapping Works

The Oracle Enterprise Session Border Controller (E-SBC) allocates SIP port mapping (signaling) ports during a REGISTER request that has registration caching applied. When you define a range of signaling ports for the SIP interface, you create a pool of signaling ports that can be allocated during the REGISTER request.

The E-SBC allocates a signaling port from the pool when it creates the registration cache entry for a Contact in a REGISTER request. It allocates a separate signaling port for each unique Contact URI from the access side. The registration cache Contact entry contains the mapping between the Contact URI in the access/endpoint realm (the UA-Contact) and the Contact URI in the registrar/softswitch realm (the SD-Contact).

The SD-Contact is the allocated signaling port. The signaling port gets returned to the pool when the Contact is removed from the registration cache. The removal can occur when the cache entry expires; or when the endpoint sends a REGISTER request to explicitly remove the Contact from the registrar. When a signaling port returns to the pool it gets placed at the end of pool list; in a least-recently-used allocation method for signaling ports

When the E-SBC forwards the REGISTER request to the softswitch, it replaces the UA-Contact with SD-Contact. For example, if UA1 sends a REGISTER request with a Contact URI of sip:ua1@172.16.0.91:5060, it is replaced with sip:192.168.24.15:2001 when the REGISTER request is forwarded to the registrar.

The same translation occurs when UA1 sends that same URI in the Contact header of other SIP messages. SIP requests addressed to the allocated signaling transport address (SD-Contact) are translated and forwarded to the registered endpoint contact address (UA-Contact).

Note: The maximum number of registered endpoints cannot exceed the number of signaling ports available. If no signaling ports are available for a new registration, the REGISTER request receives a 503 response.

The E-SBC still processes requests received on the configured SIP port address. Requests sent into the registrar/softswitch realm that are not associated with a registered user will use the configured SIP port address.

Using SIP port mapping with SIPconnect—where unique ports are used for each registered PBX—hinders the E-SBC from routing incoming calls to the corresponding PBX because the E-SBC uses DN for the PBX's parent during registration, but the incoming INVITE from the softswitch contains the child DN in its Request URI. Thus the E-SBC cannot find a matching SBC-Contact because the username of the Request URI contains the child DN, but the username of the SBC-Contact contains the parent DN.

You can enable SIPconnect support in either the realm configuration or session agent for the SIP access network by setting the sip-connect-pbx-reg option. With this option set and the destination realm configured for port mapping, the E-SBC inserts a special search key in the registration table. Rather than adding the SD-Contact as the key as with regular (non-SIPconnect) registrations, the E-SBC strips user information and instead uses the host and port information as the registration key. The E-SBC still forwards the registration message with an intact contact username.

SIP Port Mapping Based on IP Address

Some registrars need to know that multiple contacts represent the same endpoint. The extension to this feature answers the expectation from registrars that an endpoint registering multiple AoRs will use a single core-side mapped port to show that the AoRs really represent a single endpoint.

When you enable SIP port mapping based on IP Address, the Oracle Enterprise Session Border Controller supports core-side UDP port mapping based on the endpoint's IP address. It ignores the username portion of the AoR or Contact.

The Oracle Enterprise Session Border Controller performs the port mapping allocation and lookup based on all requests using the via-key from the SIP Request. The via-key is a combination of Layer 3 and Layer 5 IP information in the message. The Oracle Enterprise Session Border Controller performs an additional lookup in the registration table to determine if a via-key already exists. If it does, then the Oracle Enterprise Session Border Controller uses the port already allocated and does not allocate a new one.

About NAT Table ACL Entries

To enable SIP signaling messages to reach the host processor, the Oracle Enterprise Session Border Controller adds NAT table ACL entries for each SIP interface. With UDP without SIP port mapping applied, it adds a single ACL entry for each SIP port in the SIP interface configuration. For example:

```
untrusted entries:
intf:vlan source-ip/mask:port/mask dest-ip/mask:port/mask   prot type    index
0/0:0     0.0.0.0                  172.16.1.15:5060         UDP  static  10
0/3:0     0.0.0.0                  192.168.24.15:5060       UDP  static  16
0/1:0     0.0.0.0                  192.168.50.25:5060       UDP  static  17
```

Using SIP Port Mapping

When you use SIP port mapping, one or more ACL entries are added to the NAT table to enable the range of ports defined. The NAT table does not support the specification of port ranges. However, it does support masking the port to enable ranges that fall on bit boundaries. For example, an entry for 192.168.24.15:4096/4 defines the port range of 4096 through 8191.

The algorithm for determining the set of ACLs for the port map range balances the need to represent the range as closely as possible, with the need to minimize the number of ACL entries. For example, a range of 30000 through 39999 would result in the following set of ACLs.

```
untrusted entries:
intf:vlan source-ip/mask:port/mask dest-ip/mask:port/mask   prot type    index
0/3:0     0.0.0.0                  192.168.24.15:30000/4    UDP  static  13
0/3:0     0.0.0.0                  192.168.24.15:32768/4    UDP  static  14
0/3:0     0.0.0.0                  192.168.24.15:36864/4    UDP  static  15
```

However, the first entry actually enables ports 28672 though 32767 and the last entry allows port 36864 through 40959. If SIP messages are received on ports outside the configured range (28672 through 29999 or 40000 through 40959 in this case), they are ignored.

Acme Packet recommends you use port map ranges that fall on bit boundaries to ensure the fewest possible ACL entries are created and only the configured ports are allowed by the ACLs. For example, a range of 32768 to 49151 provides for 16,384 signaling ports in a single ACL entry (192.168.24.15:32768/2).

Note: If the ACLs added for the port map range do not include the SIP port configured in the SIP interface; the normal SIP ACL entry for the SIP port is also added.

Dynamic Configuration

Dynamic configuration of SIP port mapping can cause disruption in service for existing registration cache entries; depending on the changes made to the defined port map range. If the range of mapping ports is reduced, it is possible that SIP signaling messages from the registrar/softswitch realm will no longer be sent to the host processor because of the changes in the NAT Table ACL entries.

When the range of mapping ports is changed, any signaling ports in the free signaling port pool not allocated to a registration cache entry are removed from the pool. When an allocated signaling port that is no longer part of the defined mapping port range is released, it is not returned to the pool of free steering ports.

The administrator is warned when the changed configuration is activated after the port map range of a SIP interface has been changed.

Registration Statistics

The SIP registration cache statistics include counters for free and allocated signaling ports. You can issue a show registration command to display the statistics:

```
17:36:55-190
SIP Registrations           -- Period -- -------- Lifetime --------
                  Active     High   Total       Total  PerMax    High
User Entries           4        4       0           7       4       4
Local Contacts         4        4       0           7       4       4
Free Map Ports     12284    12284       0       12291   12288   12288
Used Map Ports         4        4       0           7       4       4
Forwards               -        -       1          22       4
Refreshes              -        -       3          43       3
Rejects                -        -       0           0       0
Timeouts               -        -       0           1       1
Fwd Postponed          -        -       0           0       0
Fwd Rejected           -        -       0           0       0
Refr Extension         0        0       0           0       0       0
Refresh Extended       -        -       0           0       0
```

The labels for the first two items reflect the restructured registration cache:

- User Entries: counts the number of unique SIP addresses of record in the cache. Each unique address of record represents a SIP user (or subscriber). The address of record is taken from the To header in the REGISTER request. There might be one or more registered contacts for each SIP user. The contacts come from the Contact header of the REGISTER request.
- Local Contacts: counts the number of contact entries in the cache. Because the same user can register from multiple endpoints (user agents); the number of Local Contacts might be higher than the number of User Entries.
- Free Map Ports: counts the number of ports available in the free signaling port pool.
- Used Map Ports: counts the number of signaling ports allocated for registration cache entries. The value of Used Map Ports will equal the number of Local Contacts when the port mapping feature is used for all registrar/softswitch realms in the Oracle Enterprise Session Border Controller.

SIP Port Mapping Configuration

You configure the SIP port mapping feature on a per-realm basis using the SIP interface configuration. Configure the port map range on the SIP interface for the realm where the registrar/softswitch resides. Port mapping is only applied when the access/ingress realm has registration caching and/or HNT enabled.

The range of SIP mapping ports must not overlap the following:

- Configured SIP port, which might be used for signaling messages not associated with a registered endpoint.
- Port range defined for steering pool configuration using the same IP address as the SIP interface. If overlap occurs, the NAT table entry for the steering port used in a call prevents SIP messages from reaching the host processor.

To configure SIP port mapping:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. port-map-start—Set the starting port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. The default values is 0 and when this value is set, SIP port mapping is disabled. The valid range is:

   - Minimum: 0, 1025
   - Maximum: 65535

```
ACMEPACKET(sip-interface)# port-map-start 32768
```

5. port-map-end—Set the ending port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. If you set the value to the default 0, SIP port mapping is disabled. The valid range is:

   - Minimum—0, 1025
   - Maximum—65535

   Note: If not set to zero (0), the ending port must be greater than the starting port.

```
ACMEPACKET(sip-interface)# port-map-end 40959
```

6. options—If you want to use SIP port mapping based on IP address, set the options parameter by typing options, a Space, the option name reg-via-key with a plus sign in front of it, type the equal sign and the word all. Then press Enter.

```
ACMEPACKET(sip-interface)# options +reg-via-key=all
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

7. Save your work using the ACLI done command.

   The following example shows SIP port mapping configured for a SIP interface:

```
sip-interface
        state                           enabled
        realm-id                        backbone
        sip-port
                address                         192.168.24.15
                port                            5060
                transport-protocol              UDP
```

```
            allow-anonymous                 all
        sip-port
            address                         192.168.24.15
            port                            5060
            transport-protocol              TCP
            allow-anonymous                 all
        carriers
        proxy-mode
        redirect-action
        contact-mode                    none
        nat-traversal                   none
        nat-interval                    30
        registration-caching            enabled
        min-reg-expire                  120
        registration-interval           3600
        route-to-registrar              enabled
        teluri-scheme                   disabled
        uri-fqdn-domain
        trust-mode                      agents-only
        max-nat-interval                3600
        nat-int-increment               10
        nat-test-increment              30
        sip-dynamic-hnt                 disabled
        stop-recurse                    401,407
        port-map-start                  32768
        port-map-end                    40959
        last-modified-date              2005-09-23 14:32:15
```

SIP Port Mapping for TCP and TLS

In releases prior to S-C6.2.0, the Oracle Enterprise Session Border Controller (E-SBC) supports SIP port mapping for UDP and now you can enable this feature for SIP sessions using TCP and TLS. Port mapping enables the E-SBC to allocate a unique port number for each endpoint registering through it by giving it a transport address (or hostport) in the registered Contact.

When you enable this feature for TCP and TLS, the E-SBC designates a port from a configured range for each endpoint that registers with SIP servers in the SIP interface's realm. You establish that range of ports using the port-map-start and port-map-end parameters. Unlike its behavior with UDP port mapping—where the E-SBC sends requests on the SIP interface from the allocated port mapping, the E-SBC sends all requests over an existing connection to the target next hop for TCP/TLS port mapping. If a connection does not exist, the system creates one. So for TCP/TLS port mapping, only the Contact header contains the transport address of the mapping port (i.e., the transport address of the configured SIP port). And the system refuses TCP and TLS connections on the allocated mapping port.

With TCP/TLS port mapping enabled, the E-SBC sends the Path header with the transport address in Register requests, unless you specify that it should not do so. Standards-conformant SIP servers (that support RFC 3327) might attempt to send requests to the allocated mapping port if the Path header is absent.

Note: ACL entries in the NAT table that permit TCP/TLS signaling for a SIP port configuration with TCP/TLS port mapping are the same as they would be for a TCP/TLS SIP port without port mapping enabled. Additional ACL entries that need to be set up for UDP port mapping are not required for TCP/TLS port mapping.

RTN 1684

SIP Port Mapping Configuration for TCP TLS

You enable TCP/TLS port mapping in a per-realm basis using the SIP interface configuration; setting the tcp-port-mapping value in the options parameter enables the feature. Enabling this parameter turns on the port mapping feature for UDP as well.

By default, the Oracle Enterprise Session Border Controller includes the Path header in Register requests it sends from that SIP interface. If you do not this header to be included, however, you can set the value as tcp-port-mapping=nopath.

To enable TCP/TLS port mapping for a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. options—Set the options parameter by typing options, a Space, the option name tcp-port-mapping with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-interface)# options +tcp-port-mapping
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save your work.

SIP Configurable Route Recursion

When the Oracle Enterprise Session Border Controller routes SIP requests from a UAC to a UAS, it might determine that there are multiple routes to try based on a matching local policy. The Oracle Enterprise Session Border Controller recurses through the list of routes in a specific order according to your configuration and the quality of the match. There are other scenarios when a UAS replies with a 3xx Redirect response to the Oracle Enterprise Session Border Controller, the 3xx response can include multiple Contacts to which the request should be forwarded in a specific order. In both cases, the Oracle Enterprise Session Border Controller needs to recurse through a list of targets.

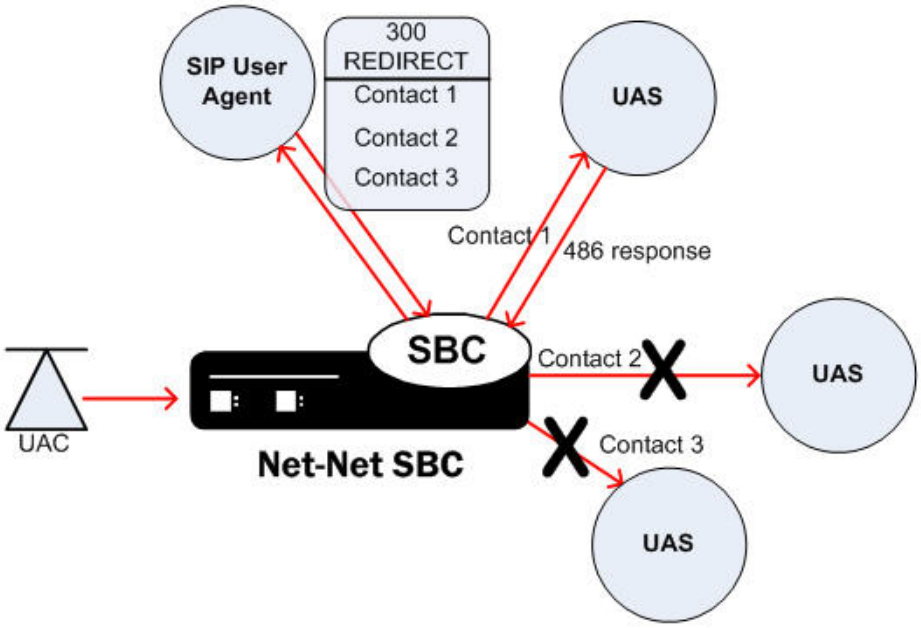
When the Oracle Enterprise Session Border Controller receives a non-successful (or non-6xx response) final response from the UAS, and there are multiple targets for the original request, the Oracle Enterprise Session Border Controller will forward the request to the next target and wait for a response. While the process of forwarding the request to multiple targets as explained in the previous paragraph is called serial forking, and the process of forwarding the request to contacts received in redirect responses is called recursion, the term recursion is used for both processes in this notice.

Use the SIP Route Recursion feature when you want the Oracle Enterprise Session Border Controller to forward a response to the UAC and stop recursing through the target list immediately after receiving the 3xx, 4xx, or 5xx response code that you configure. When this feature is disabled, the Oracle Enterprise Session Border Controller only stops recursing when it receives a message with a 401 or 407 response code. Using this feature, you can configure a specific message or range of messages to stop recursing on when received. The Oracle Enterprise Session Border Controller retains its default behavior to stop recursing on a 401 or 407 response code when SIP Route Recursion is configured on a SIP interface. The Oracle Enterprise Session Border Controller will always stop recursing when it receives a global failure (6xx); this behavior is not configurable.

You can disable response recursion for either a SIP interface or for a SIP session agent, providing you with flexibility for various network architectures. For instance, a PSTN gateway might be the only hop to reach a given endpoint, whereas several session agents might need to be contacted if multiple devices map to a contacted address of record.

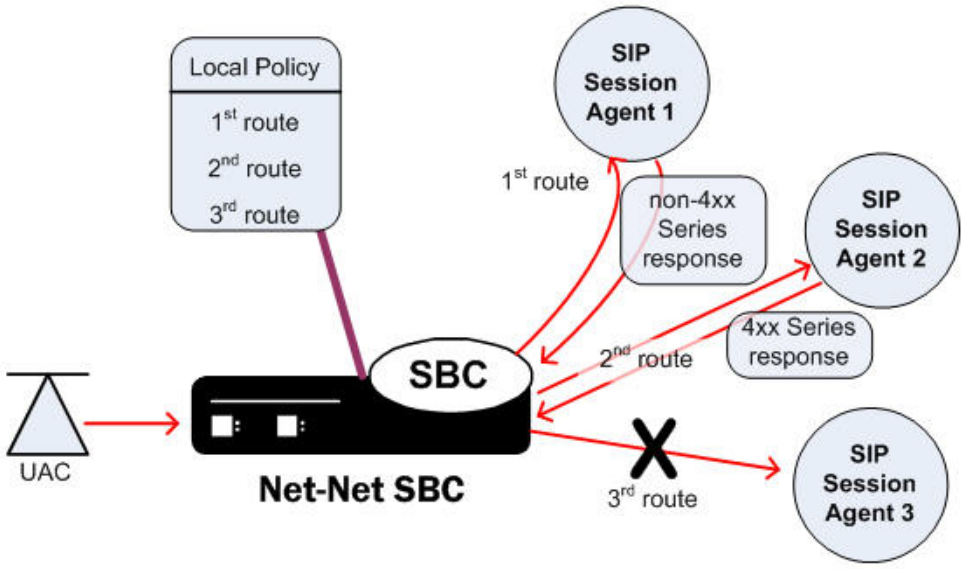
Example 1

A more detailed example is when a softswitch might return a list of contacts for multiple PSTN gateways in a Redirect message. If the PSTN target number contacted on redirection is busy, a 486 response will be sent to the Oracle Enterprise Session Border Controller. Since the single target is located in the PSTN, a subsequent request through a different gateway will yield another 486 response. The Oracle Enterprise Session Border Controller should be configured to return the 486 response to the UAC immediately. No other SIP requests should be sent to applicable targets/contacts that were enumerated in the redirect list. See the following example:



Example 2

The Oracle Enterprise Session Border Controller might determine from a local policy lookup that several routes are applicable for forwarding a SIP message. The Oracle Enterprise Session Border Controller will try each route in turn, but the SIP response recursion disable feature can be implemented to stop the route recursion when a configured responses message is received by the Oracle Enterprise Session Border Controller. See the following example:



There are a few conditions on the parameter used to configure response recursion:

- SIP Route Recursion is configurable for either the SIP interface or session agent.
- 401 and 407 are preconfigured for all configured SIP interfaces. They are not configured for session agents.
- The format is a comma-separated list of response codes or response code ranges: 404, 484-486.
- Only response codes that fall within the 3xx, 4xx, and 5xx range may be specified.

SIP Route Recursion Configuration

You enable SIP route recursion either in the session agent or the SIP interface configuration.

Configuring a Session Agent for SIP Route Recursion

To configure SIP Route recursion for an existing session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Select the session agent where you want this feature.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: asd          realm=      ip=1.0.0.0
2: SIPSA        realm=      ip=10.10.102.1
selection:2
ACMEPACKET(session-agent)#
```

5. stop-recurse—Enter list of returned response codes that this session agent will watch for in order to stop recursion on the target's or contact's messages. This can be a comma-separated list or response code ranges.

```
ACMEPACKET(session-agent)# stop-recurse 404,484-486
```

6. Save and activate your changes.

Configuring a SIP Interface for SIP Route Recursion

To configure SIP route recursion for an existing SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Select the SIP interface to which you want to apply this feature.

```
ACMEPACKET(sip-interface)# select
<realm-id>:
1: Acme_Realm
selection:1
ACMEPACKET(sip-interface)#
```

5. stop-recurse—Enter a list of returned response codes that this SIP interface will watch for in order to stop recursion on the target's or contact's messages. This list can be a comma-separated list of response codes or response code ranges.

```
ACMEPACKET(sip-interface)# stop-recurse 404,484-486
```

6. Save and activate your changes.

SIP Event Package Interoperability

Service providers often deploy a Oracle Enterprise Session Border Controller on the border of an access network, where it sits between the SIP endpoints (user agents) and the service provider's application server. The application server and the user agents sometimes use various SIP event packages to exchange and maintain state information. The SUBSCRIBE and NOTIFY methods are used to establish subscriptions to the event packages and to report state changes to the subscribing entity.

The SIP global contact option addresses interoperability in the Dialog and Presence event packages that are used in hosted PBX and IP Centrex offerings. State information is passed in the message body of a NOTIFY request; this message body is encoded in an XML format described by the Content-Type header. The Oracle Enterprise Session Border Controller needs to update certain fields in the body to account for dialog mapping and SIP NAT functionality between the access and service provider realms. Often the subscriptions are established using URIs learned from Contact headers in the user agent registrations or dialog establishment (INVITE/SUBSCRIBE). For this, a Oracle Enterprise Session Border Controllerrequires a Contact URI that is usable and routable outside of an existing dialog.

The SIP global contact option enables persistent URIs in the Contact headers inserted into outgoing SIP messages. If this option is not used, URIs placed in the Contact header of outgoing messages are only valid within the context of the dialog to which the message is associated.

RFCs associated with this feature are:

- A. B. Roach, *Session Initiation Protocol (SIP)-Specific Event Notification*, RFC 3265, June 2002
- J. Rosenberg, *A Presence Event Package for the Session Initiation Protocol (SIP)*, RFC 3856, August 2004
- J. Rosenberg, et al. *Data Format for Presence Using XML*, http://www.iptel.org/info/players/ietf/presence/outdated/draft-rosenberg-impp-pidf-00.txt, Work In Progress (expired), June 2000
- J.Rosenberg, H. Schulzrinne, R. Mahy, *An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)*, draft-ietf-sipping-dialog-package-06.txt, Work In Progress, April 2005
- H. Sugano, et al., *Presence Information Data Format (PIDF)*, RFC 3863, August 2004

SIP Event Package Interoperability Configuration

This feature is applicable to the global SIP configuration.

To configure SIP event package interoperability:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Add SIP event package interoperability support to a new SIP configuration or to an existing SIP configuration:

   If you do not currently have an SIP configuration, you can add the option by typing options, a Space and then global-contact.

```
ACMEPACKET(sip-config)# options global-contact
```

   Select the SIP configuration so that you can add SIP event package interoperability support to it. Then, to add this option to a list of options that you have already configured, type options followed by a Space, the plus sign (+), and the global-contact option.

```
ACMEPACKET(sip-config)# select
ACMEPACKET(sip-config)# options +global-contact
```

   If you type options global-contact without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your changes.

SIP Proxy Subscriptions

When the Oracle Enterprise Session Border Controller operates in dialog mode (i.e., as a B2BUA), it creates and maintains dialog state for subscription dialogs created with SUBSCRIBE/NOTIFY messages and for INVITE-initiated dialogs. Since there can be a very large number of subscriptions per user in a Rich Communication Services (RCS) environment (especially for presence subscriptions), Oracle Enterprise Session Border Controller resources can quickly become depleted.

To alleviate this consumption of resources, you can configure your Oracle Enterprise Session Border Controller to operate in proxy mode for event packages that you define using the proxy-sub-event parameter in the global SIP configuration. When you define event packages in this list and the operation mode for the SIP configuration is dialog or session, the Oracle Enterprise Session Border Controller processes all SUBSCRIBE and NOTIFY requests and responses for the designated event packages in transaction stateful mode.

Topology Hiding

So that it can perform topology hiding, the Oracle Enterprise Session Border Controller retains necessary routing information (such as the Contact or Record-Route header values) and it encodes certain data from the messages it receives in the messages it sends. To be more specific, the Oracle Enterprise Session Border Controller encodes the original URI hostport and the ingress realm name into a gr parameter it adds to the URI. The hostport information is replaced with the IP address and port of the SIP interface from which the message is sent (the egress interface). Without this information, subsequent in-dialog messages cannot be routed correctly because the Oracle Enterprise Session Border Controller does not retain dialog state (i.e., the route-set or remote-target).

For example, the URI sip:td@192.168.24.121:5060 might be encoded as sip:td@192.168.24.121:5060;gr=vjml9qtd175bhmhvhkgp0jov81popvbp000040.

The Oracle Enterprise Session Border Controller also performs URI encoding on the message body for Content-Type application/pdif+xml. this contains a Presence Information Data Format document (or PIDF) in PUBLISH and NOTIFY requests so subsequent SIP requests using the URIs in the PIDF document can be routed correctly.

The Oracle Enterprise Session Border Controller performs URI encoding in outgoing SIP messages after SIP=NAT is applied and before outbound HMR occurs. And the system decodes URIs in SIP messages after inbound HMR takes place and before SIP-NAT is applied.

In the event a URI is encoded several times (as is the case in spiral and hairpin calls), the encoded realm+hostport values are separated by a plus sign (+), as in the following:

```
sip:td@192.168.24.121:5060; gr=vjml9qtd175bhmhvhkgp+dhfhb0jov81opvbp
```

When the Oracle Enterprise Session Border Controller receives any of the following requests, it matches the contents of the request's Event header with the list you configure in the proxy-sub-events parameter:

- PUBLISH
- SUBSCRIBE
- NOTIFY
- REFER

This is provided the operation-mode for the SIP configuration is set to either session or dialog. If it finds a match, the Oracle Enterprise Session Border Controller marks the request for processing in transaction-stateful mode rather than in B2BUA mode.

Note: Although PUBLISH is not a dialog-creating request, topology hiding needs to be applied to the PIDF so that subsequent NOTIFY requests containing portions of the published PIDF can be decoded properly.

When the Oracle Enterprise Session Border Controller forwards the request, it will have encoded all Contact and Record-Route header information using the ingress realm. The hostport value of the URIs then has egress SIP interface's IP address and port. The Via headers in the requested the Oracle Enterprise Session Border Controller received are not included in the outgoing request.

Then PUBLISH, SUBSCRIBE, NOTIFY, and REFER responses are compared to the request that was sent to determine if the response should receive transaction-stateful proxy treatment. The Oracle Enterprise Session Border Controller decodes any encoded Record-Route headers back to their original values for the outgoing response. Any Record-Route headers added downstream from the Oracle Enterprise Session Border Controller are encoded using the original request's egress realm (meaning the realm from which the response was received). In addition, the Contact header is encoded using the request's egress realm and ingress SIP interface and port.

Feature Interaction

When using this feature, the Oracle Enterprise Session Border Controller does not keep dialog or subscription state. Therefore the Per-user SUBSCRIBE Dialog Limit feature—configured in the enforcement-profile configuration—will not function properly when a subscription is handled in proxy mode.

SIP Proxy Subscription Configuration

This section shows you how to configure a list of SIP event packages to cause the Oracle Enterprise Session Border Controller to act in proxy mode.

Note: The operation-mode parameter for the global SIP configuration must be set to either dialog or session in order for this feature to function as designed.

To configure a list of SIP event packages to enable SIP proxy subscriptions:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration that you want to edit.

4. proxy-sub-events—Enter a list of SIP event package names that you want to enable the SIP proxy subscriptions feature. You can enter more than one value by enclosing multiple values in quotations marks, as in the following example.

```
ACMEPACKET(sip-config)# proxy-sub-events presence winfo
```

SIP REGISTER Forwarding After Call-ID Change

This feature addresses the case when an endpoint reboots and performs a third party registration before its old registration expires. During this reregistration, the contact header is the same as it was pre-reregistration. As a consequence of the reboot, the SIP Call-ID changes. In this situation, the Oracle Enterprise Session Border Controller does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI.

To remedy this problem, the Oracle Enterprise Session Border Controller now keeps track of the Call-ID in its registration cache. The forward-reg-callid-change option in the global SIP configuration element forces the Oracle Enterprise Session Border Controller to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.

SIP REGISTER Forwarding Configuration

To configure SIP REGISTER forwarding after a Call-ID change:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Add this feature to a new or an existing SIP configuration:

If you do not currently have a SIP configuration, you can add the option by typing options, a Space, and then forward-reg-callid-change.

```
ACMEPACKET(sip-config)# options forward-reg-callid-change
```

For an existing SIP configuration, select the SIP configuration so that you can add this feature to it. Then, to add this option to a list of options that you have already configured, type options, a Space, the plus sign (+), and the forward-reg-callid-change option.

```
ACMEPACKET(sip-config)# options +forward-reg-callid-change
```

If you type options forward-reg-callid-change without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your changes.

SIP Local Response Code Mapping

The SIP local response code mapping feature has been added as an enhancement to the SIP response code mapping. The SIP response code map feature lets you establish a table that maps SIP response-received messages (entries) to response-to-send messages (entries).

SIP local response code mapping is used with the SIP responses generated by the Oracle Enterprise Session Border Controller towards a specific SIP session agent. This feature lets you provision the mapping of the response codes used by the Oracle Enterprise Session Border Controller when it generates the responses towards a session agent.

You create the SIP local response code map using the existing mapping functionality, and then assigning that map to a session agent or to a SIP interface.

Note: The configured response map is not used when the Oracle Enterprise Session Border Controller is acting as proxy for the responses to this session agent.

SIP Local Response Code Mapping Configuration

The following instructions explain how to create the SIP response code map and then how to assign it to a specific session agent.

Creating a SIP Response Code Map

To create a SIP local response code map:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-response-map and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-response-map
ACMEPACKET(response-map)#
```

4. name—Enter the name of the SIP response map you want to configure. This value is required and must be unique.

```
ACMEPACKET(response-map)# name busy
```

5. entries—To configure the entries for this mapping, type entries and then press Enter. Typing a question mark will show you the response code entry parameters that you can configure.

```
ACMEPACKET(response-map)# entries
ACMEPACKET(response-map-entries)#
```

recv-code—Enter original SIP response code for the recv-mode parameter. The valid range is:

- Minimum—100
- Maximum—699

```
ACMEPACKET(response-map-entries)# recv-mode 486
```

xmit-code—Enter the SIP response code into which you want the original response code to be translated. This valid range is:

- Minimum—100
- Maximum—699

```
ACMEPACKET(response-map-entries)# xmit-mode 600
```

reason—Enter a reason for the translated code into the reason parameter. This response comment is sent with the translated code. Make your entry in quotation marks.

```
ACMEPACKET(response-map-entries)# reason Busy Everywhere
```

The following two parameters (method and register-response-expires) enable a SIP registration response mapping feature that allows you to configure the Oracle Enterprise Session Border Controller to remap a SIP failure response—which it receives from another network device or that it generates locally—to a 200 OK. You might want the Oracle Enterprise Session Border Controller to perform this type of mapping for circumstances where non-malicious endpoints continually attempt registration, but will stop (and still not be registered) when they receive a 200 OK. This response mapping does not actually register the client with the Oracle Enterprise Session Border Controller, meaning that there is neither a registration cache entry or a CAM ACL for it.

For the 200 OK it generates, the Oracle Enterprise Session Border Controller removes any Reason or Retry-After header in the 200 OK and sets the expires time. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). You can also set this value using the register-response-expires parameter, but the value you set should never exceed the Register request's expires time.

method—Enter the name of the received SIP failure response message you want to map to a 200 OK. There is no default for this parameter, and leaving the parameter empty turns off the SIP registration response mapping feature.

register-response-expires—Enter the time you want to use for the expires time what mapping the SIP method you identified in the method parameter from Step 4. The maximum is 999999999. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). Any value you configure in this parameter (when not using the defaults) should never exceed the Register request's expires time.

6. Note the name that you gave the SIP response code map so that you can use it when you configure a session agent to support SIP response code mapping.
7. Save and activate your changes.

Assigning SIP Response Code Maps to Session Agents

To assign a SIP local response code map to a session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. local-response-map—Enter the name of the configured SIP response map that you want to use for this session-agent and press Enter.

```
ACMEPACKET(session-agent)# local-response-map busy
```

5. Save and activate your configuration.

Assigning SIP Response Code Maps to SIP Interfaces

To apply SIP response codes maps to a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. local-response-map—Enter the name of the configured SIP response map that you want to apply to this SIP interface for locally-generated SIP responses. This parameter is blank by default.
5. Save and activate your configuration.

Session Agent Ping Message Formatting

You can configure the user portions of the Request-URI and To: headers that define the destination of a session agent ping message, and the From: header that defines the source of a session agent ping message. These headers are sent to Oracle Enterprise Session Border Controller session agent. This feature is required for interoperability with certain E911 servers.

In the following example of a session agent ping-type message, you can set the user portion of the Request-URI (the text bob in the OPTIONS method line) and the user portion of the From: header (the text bob in the From: header) to the same new value. You can also set the user portion of the To: header (the text anna in the To: header) to its own new value.

```
OPTIONS sip:bob@sip.com SIP/2.0
From: UA1 <sip:bob@sip.com>
To: NUT <sip:anna@gw.sip.com>
Call-ID: 123abc@desk.sip.com
CSeq: 1 OPTIONS
Contact: <sip:UA1@client.sip.com>
Accept: application/sdp
Content-Length: 0
```

If you do not enable this feature, then the session agent ping-type message contains the text ping in all cases.

Session Agent Ping Message Formatting Configuration

1. Access the **session-agent** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)
```

2. Select the **session-agent** object to edit.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: 192.168.100.101:1813

selection: 1
ACMEPACKET(session-agent)#
```

3. **ping-from-user-part**—Set the user portion of the From: header that defines the source of a session agent ping message.

```
ACMEPACKET(session-agent)# ping-from-user-part bob
```

4. **ping-to-user-part**—Set the user portions of the Request-URI and To: headers that define the destination of a session agent ping message.

```
ACMEPACKET(session-agent)# ping-to-user-part anna
```

5. Type **done** to save your configuration.

SIP PAI Stripping

The Oracle Enterprise Session Border Controller now has the ability to strip P-Asserted-Identity (PAI) headers so that service providers can ensure an extra measure of security against malicious users pretending to be legitimate users. To

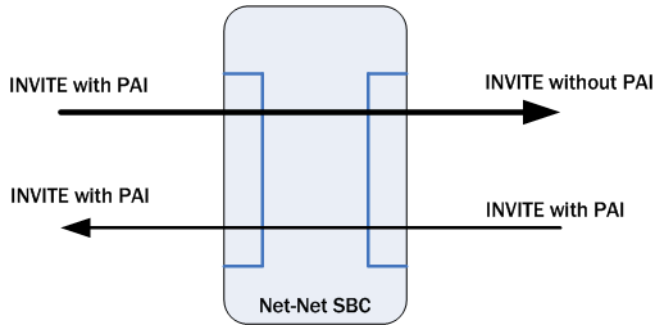
pretend to represent another account, the malicious users simply send an INVITE with an imitation PAI. This feature allows real-time detection of such fraudulent use.

This feature uses a combination of:

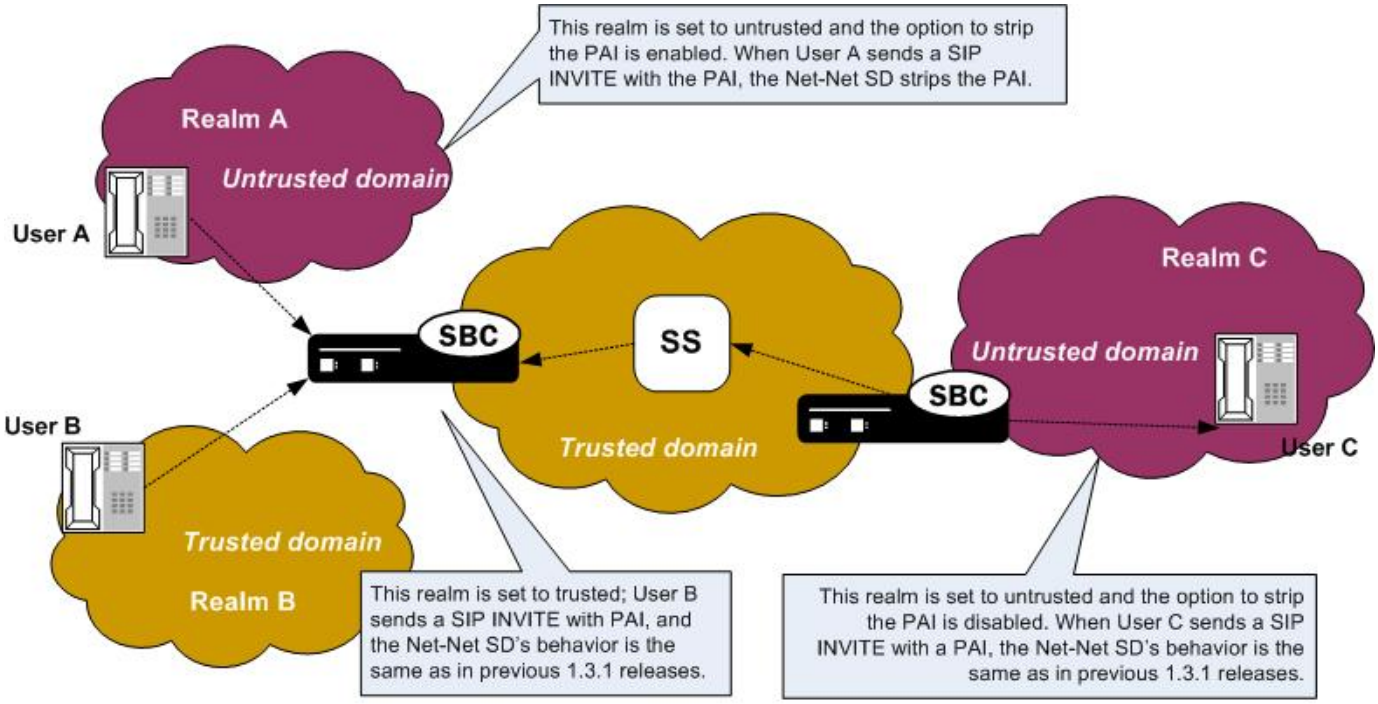
- DoS protection applied on a per-realm basis
- SIP PAI header stripping

The combination of these settings can produce different results for the SIP PAI stripping feature.

- SIP PAI header stripping enabled for an untrusted realm—If the PAI stripping parameter is set to enabled in a realm that is untrusted, then the Oracle Enterprise Session Border Controller strips the PAI headers from SIP INVITEs that are received from the external address, regardless of the privacy type. The Oracle Enterprise Session Border Controller then sends the modified INVITE (without the PAI). If the INVITE comes from a trusted realm, then the Oracle Enterprise Session Border Controller does not strip the PAI header and the system behaves as it does when you are using previous 1.3.1 releases.



- Multiple SIP PAIs in a SIP INVITE—The Oracle Enterprise Session Border Controller removes all PAIs when there are multiple PAIs set in SIP INVITEs that come from untrusted realms.
- Oracle Enterprise Session Border Controller behavior bridging trusted and untrusted realms—The following graphics shows you how Oracle Enterprise Session Border Controllers can be positioned and configured to handle PAI stripping between trusted and untrusted realms.

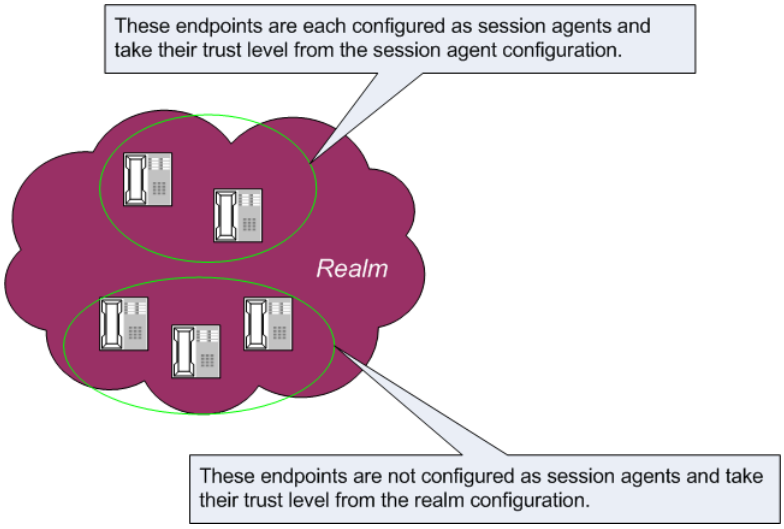


Realm Configuration Settings	REALM A	REALM B	REALM C
Realm designation trusted or untrusted (trust-me)	Disabled	Enabled	Enabled
SIP PAI stripping (pai-strip)	Enabled	Enabled or disabled	Disabled
SBC's behavior	Strip PAI regardless of privacy type	Same as behavior for SIP privacy support in previous 1.3.1 releases	Same as behavior for SIP privacy support in previous 1.3.1 releases

SIP PAI Stripping Configuration

When you configure this feature, please note how the Oracle Enterprise Session Border Controller behaves when you combine the designation of a realm as trusted/untrusted and SIP PAI stripping is enabled. Enter the choices for the ACLI trust-me and pai-strip parameters accordingly.

Be aware that trust is also established in the session agent configuration, and that the trust level set in a session agent configuration overrides the trust set in a realm configuration. For example, a realm might have several endpoints, some of which are associated with session agents and some of which are not. The endpoints that have configured session agent will take their trust level from the session agent parameters you set; the other endpoints, ones that are not associated with session agents, take their trust level from the realm parameters you set.



Take this relationship into consideration when you configure SIP PAI header stripping, or this feature will not work as designed.

For the sample configuration cited below, the desired Oracle Enterprise Session Border Controller behavior is to always strip the PAI regardless of privacy type.

To configure SIP PAI stripping for an existing realm using the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm to which you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: acmePacket <none>           192.168.20.0/24
2: realm1     <none>           0.0.0.0
selection:2
ACMEPACKET(realm-config)#
```

5. pai-strip—Enable PAI stripping. The default is disabled. Valid values are:

   - enabled | disabled

```
ACMEPACKET(realm-config)# pai-strip enabled
```

6. Save your work using the ACLI save or done command.

SIP Statuses to Q.850 Reasons

This section explains the Oracle Enterprise Session Border Controller's ability to map Q.850 cause values with SIP responses, a feature used in SIP calls and calls that require IWF.

RFC 3326 defines a header that might be included in any in-dialogue request. This reason header includes cause values that are defined as either a SIP response code or ITU-T Q.850 cause values. You can configure the Oracle Enterprise Session Border Controller to support sending and receiving RFC 3326 in SIP messages for:

- Mapping H.323 Q.850 cause values to SIP responses with reason header and cause value

- Mapping SIP response messages and RFC 3326 reason header and cause
- Locally generated SIP response with RFC 3326 reason header and cause

As specified in RFC 3326, the Oracle Enterprise Session Border Controller sends SIP responses to the softswitch that contain the received Q.850 cause code and the reason.

Though the Oracle Enterprise Session Border Controller can generate RFC 3326 headers, the default behavior for this feature is disabled. Furthermore, the Oracle Enterprise Session Border Controller can receive and pass SIP error messages (4xx, 5xx, and 6xx) that contain the SIP reason header with a Q.850 cause code and reason (as specified in RFC 3326). If the system receives an error message without the Reason header, then the Oracle Enterprise Session Border Controller is not required to insert one.

In calls that require IWF, the Q.850 cause generated in the SIP response are the same as the cause received in the following H.225 messages: Disconnect, Progress, Release, Release Complete, Resume Reject, Status, and Suspend Reject. In addition, the Q.850 cause codes that the Oracle Enterprise Session Border Controller receives in RFC 3326 headers are passed to the H.323 part of the call unmodified; the H.323 call leg uses this cause code for releasing the call.

SIP-SIP Calls

The SIP Reason header might appear in any request within a dialog, in a CANCEL request, and in any response where the status code explicitly allows the presence of this header field. The syntax of the header follows the standard SIP parameter:

```
Reason: SIP;cause=200;text=”completed elsewhere”
Reason: Q.850;cause=16;text=”Terminated”
```

This feature attends to the following possible SIP call scenarios:

- When the Oracle Enterprise Session Border Controller receives a SIP request or SIP response that contains the Reason header, the Oracle Enterprise Session Border Controller passes it without modification.
- When it generates a SIP response, the Oracle Enterprise Session Border Controller includes the RFC 3326 Reason header containing a Q.850 cause code and reason. This is the case for all local conditions and for all internally generated error responses (4xx, 5xx, and 6xx) to an initial SIP INVITE.

  Possible local error scenarios are:

  - invalid-message
  - cpu-overloaded
  - media-released
  - media-not-allocated

SIP-SIP Calls Configuration

Configuring reason cause mapping for SIP-SIP calls requires that you set up the ACLI local-response-map configuration with appropriate entries; these generate the SIP response and the Q.850 cause code value to be used for particular error scenarios. If you want to add a Reason header, then you need to enable that capability in the global SIP configuration.

To configure a local response map:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type local-response-map and press Enter.

   ```
   ACMEPACKET(session-router)# local-response-map
   ACMEPACKET(local-response-map)#
   ```

4. Type entries and press Enter.

```
ACMEPACKET(local-response-map)# entries
ACMEPACKET(local-response-map-entry)#
```

From here, you can view the entire menu for the local response map entries configuration by typing a ?.

5. local-error—Set the local error that triggers the use of this local response map; there is no default for this parameter. Valid values are:
   - invalid-message—Response map for invalid messages
   - cpu-overload—Response map for CPU overload
   - enum-void-route—Response map for when an ENUM server returns a ENUM+VOID response, or the local route table has 0.0.0.0 as the next hop
   - media-released—Response map for media release conditions
   - media-not-allocated—Response map for when media is not allocated
6. sip-status—Set the SIP response code to use. There is no default and the valid range is:
   - Minimum—100
   - Maximum—699
7. sip-reason—Set the SIP reason string you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. q850-cause—Set the Q.850 cause. There is no default value.
9. q850-reason—Set the Q.850 reason string that you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.

   Note:

   The following two parameters (method and register-response-expires) enable a SIP registration response mapping feature that allows you to configure the system to remap a SIP failure response—which it receives from another network device or that it generates locally—to a 200 OK. You might want the system to perform this type of mapping for circumstances where non-malicious endpoints continually attempt registration, but will stop (and still not be registered) when they receive a 200 OK. This response mapping does not actually register the client with the system, meaning that there is neither a registration cache entry or a CAM ACL for it.

   For the 200 OK it generates, the system removes any Reason or Retry-After header in the 200 OK and sets the expires time. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). You can also set this value using the register-response-expires parameter, but the value you set should never exceed the Register request's expires time.
10. method—Enter the name of the received SIP failure response message you want to map to a 200 OK.

    There is no default for this parameter, and leaving the parameter empty turns off the SIP registration response mapping feature.
11. register-response-expires—Enter the time you want to use for the expires time what mapping the SIP method you identified in the method parameter from Step 4.

    The maximum is 999999999. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). Any value you configure in this parameter (when not using the defaults) should never exceed the Register request's expires time.
12. Repeat this process to create the number of local response map entries that you need.
13. Save and activate your configuration for changes to take effect.

Adding the Reason Header

To enable the Oracle Enterprise Session Border Controller to add the Reason header:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. add-reason-header—Enable this parameter to add the Reason header.

   The default value is disabled. The valid values are:

   - enabled | disabled

5. Save and activate your configuration for changes to take effect.

Calls Requiring IWF

For interworking calls between SIP and H.323, you can configure:

- Mappings for SIP status codes to Q.850 values
- Mappings for particular Q.850 cause codes to SIP status codes

If it cannot find the appropriate mapping, then the Oracle Enterprise Session Border Controller uses default mappings defined in the Default Mappings table below.

The following describes how the Oracle Enterprise Session Border Controller handles different IWF call scenarios:

- SIP request containing a Reason header—When it receives a request containing a Reason header, the Oracle Enterprise Session Border Controller determines if the request is a SIP BYE or SIP CANCEL message. RFC 3326 states that the Reason header is mainly used for these types of requests. If there is a Reason header and it contains the Q.850 cause value, then the Oracle Enterprise Session Border Controller releases the call on the H.323 side using the specified cause value.
- SIP response—When it receives the error response to an initial SIP INVITE, the Oracle Enterprise Session Border Controller uses its SIP-Q.850 map to determine the Q.850 that it will use to release the call. If there is not a map entry, then the Oracle Enterprise Session Border Controller uses the default mappings shown in the Default Mappings table.
- Active call released from the H.323 side—If an active call is released from the H.323 side, the Oracle Enterprise Session Border Controller checks the outgoing realm (the SIP side) to see if the addition of the Reason header is enabled. If it is, then the Oracle Enterprise Session Border Controller adds the Reason header in the SIP BYE request with the Q.850 value it received from the H.323 side.
- Error during setup of the call on the H.323 side—In the event of an error during setup on the H.323 side of the call, the system needs to send:
  - An error response, if this is a SIP to H.323 call
  - A SIP CANCEL, if this is a H.323 to SIP call and the H.323 side hangs up before the call is answered on the SIP side

    In this case, the Oracle Enterprise Session Border Controller checks to see if adding the Reason header is enabled in the IWF configuration. If it is, then the Oracle Enterprise Session Border Controller adds the Reason header with the Q.850 cause value it received from the H.323 side.
- Call released due to a Oracle Enterprise Session Border Controller error—If the call is released due a Oracle Enterprise Session Border Controller error and adding the Reason header is enabled in the IWF configuration, the error response to the initial INVITE contains the Reason header. The Oracle Enterprise Session Border Controller checks the SIP to Q.850 map configurations to determine whether or not the SIP error response code it is generating is configured. If it is, then the Oracle Enterprise Session Border Controller maps according to the configuration. If if it not, the Oracle Enterprise Session Border Controller derives cause mapping from the default table.

Like the configuration for SIP-only calls that enable this feature, you can set a parameter in the IWF configuration that enables adding the Reason header in the SIP requests or responses.

Default Mappings

This table defines the default mappings the Oracle Enterprise Session Border Controller uses when it cannot locate an appropriate entry that you have configured.

Q.850 Cause Value		SIP Status		Comments
1	Unallocated number	404	Not found	
2	No route to specified transit network	404	Not found	
3	No route destination	404	Not found	
16	Normal calling clearing		BYE message	A call clearing BYE message containing cause value 16 normally results in the sending of a SIP BYE or CANCEL request. However, if a SIP response is to be sent to the INVITE request, the default response code should be used.
17	User busy	486	Busy here	
18	No user responding	408	Request timeout	
19	No answer from the user	480	Temporarily unavailable	
20	Subscriber absent	480	Temporarily unavailable	
21	Call rejected	603	Decline (if location filed in Cause information element indicates user; otherwise 403 Forbidden is used)	
22	Number changed	301	Moved permanently (if information in diagnostic field of Cause information element is suitable for generating SIP Contact header; otherwise 410 Gone is used)	
23	Redirection to new destination	410	Gone	
25	Exchange routing error	483	Too many hops	
27	Destination out of order	502	Bad gateway	
28	Address incomplete	484	Address incomplete	
29	Facility rejected	501	Not implemented	
31	Normal, unspecified	480	Temporarily unavailable	
34	No circuit, channel unavailable	503	Service unavailable	
38	Network out of order	503	Service unavailable	

Q.850 Cause Value		SIP Status		Comments
41	Temporary failure	503	Service unavailable	
42	Switching equipment congestion	503	Service unavailable	
47	Resource unavailable unspecified	503	Service unavailable	
55	Incoming calls barred with CUG	403	Forbidden	
57	Bearer capability not authorized	403	Forbidden	
58	Bearer capability not presently available	503	Service unavailable	
65	Bearer capability not implemented	488	Not acceptable here	
69	Requested facility not implemented	501	Not implemented	
70	Only restricted digital information available	488	Not acceptable here	
79	Service or option not implemented, unspecified	501	Not implemented	
87	User not member of CUG	403	Forbidden	
88	Incompatible destination	503	Service unavailable	
102	Recovery on timer expiry	504	Server time-out	

SIP Status

To configure a SIP status to Q.850 Reason with cause mapping:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-q850-map and press Enter.

```
ACMEPACKET(session-router)# sip-q850-map
ACMEPACKET(sip-q850-map)#
```

4. Type entries and press Enter.

```
ACMEPACKET(sip-q850-map)# entries
ACMEPACKET(sip-q850-map-entry)#
```

   From here, you can view the entire menu for the SIP status to Q.850 Reason with cause mapping entries configuration by typing a ?.

5. sip-status—Set the SIP response code that you want to map to a particular Q.850 cause code and reason. There is no default, and the valid range is:

- Minimum—100
- Maximum—699

6. q850-cause—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
7. q850-reason—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

To configure a Q.850 cause to a SIP status with reason mapping:

10. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

11. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

12. Type sip-q850-map and press Enter.

```
ACMEPACKET(session-router)# q850-sip-map
ACMEPACKET(q850-sip-map)#
```

13. Type entries and press Enter.

```
ACMEPACKET(q850-sip-map)# entries
ACMEPACKET(q850-sip-map-entry)#
```

From here, you can view the entire menu for the Q.850 cause to a SIP response code with reason mapping entries configuration by typing a ?.

14. q850-cause—Set the Q.850 cause code that you want to map to a SIP status with reason. There is no default.
15. sip-status—Set the SIP response code to which you want to map the Q.850 cause that you set in step 5. There is no default, and the valid range is:

- Minimum—100
- Maximum—699

16. sip-reason—Set the reason that you want to use with the SIP response code that you specified in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
17. Repeat this process to create the number of local response map entries that you need.
18. Save and activate your configuration for changes to take effect.

To enable the Oracle Enterprise Session Border Controller to add the Reason header for calls that require IWF:

19. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

20. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

21. Type iwf-config and press Enter.

```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```

22. add-reason-header—Enable this parameter to add the Reason header. The default is disabled. The valid values are:

- enabled | disabled

23. Save and activate your configuration for changes to take effect.

Trunk Group URIs

The Oracle Enterprise Session Border Controller's trunk group URI feature, applicable for SIP and IWF signaling services, enables the capabilities related to trunk groups that are described in this section. This implementation

follows the IPTEL draft Representing Trunk Groups in Tel/SIP Uniform Resource Identifiers (URIs) (draft-ietf-iptel-trunk-group-06.txt), and also supports more customized approaches.

- For a typical access call flow scenario, when the calling party's call arrives at the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller formulates a SIP INVITE message that it sends to a softswitch. The Oracle Enterprise Session Border Controller now supports a new URI contact parameter in the SIP request message so that service providers need to be able to:
  - Determine from where the Oracle Enterprise Session Border Controller received the call
  - Signal information about the originating gateway from a Oracle Enterprise Session Border Controller to a softswitch (e.g., an incoming trunk group or a SIP gateway to a Oracle Enterprise Session Border Controller)
- This feature supports the signaling of routing information to the Oracle Enterprise Session Border Controller from network routing elements like softswitches. This information tells the Oracle Enterprise Session Border Controller what egress route (or outgoing trunk groups) it should choose for terminating next hops/gateways. For this purpose, new SIP URI parameters in the Request-URI are defined. Additional URI parameters include the network context to identify the network in which the originating or terminating gateway resides.
- Especially important for large business applications, this feature can free Oracle Enterprise Session Border Controller resources by reducing the number of local policy, session agent, and session agent group configurations. By enabling the trunk group URI feature, the Oracle Enterprise Session Border Controller instead uses a routing scheme based on signaled SIP URI information.

Terminology

The following IPTEL terms are used in the descriptions of and instructions for how to configure this feature:

- Trunk—In a network, a communication path connecting two switching systems used in the establishment of an end-to-end connection; in selected applications, it may have both its terminations in the same switching system
- Trunk group—A set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable except where sub-grouped
- Trunk group name—Provides a unique identifier of the trunk group; referred to as tgrp
- Trunk group context—Imposes a namespace by specifying a domain where the trunk groups are; also referred to simply as context

Trunk Group URI Parameters

Trunk group URI parameters identify originating and terminating trunk group information in SIP requests.

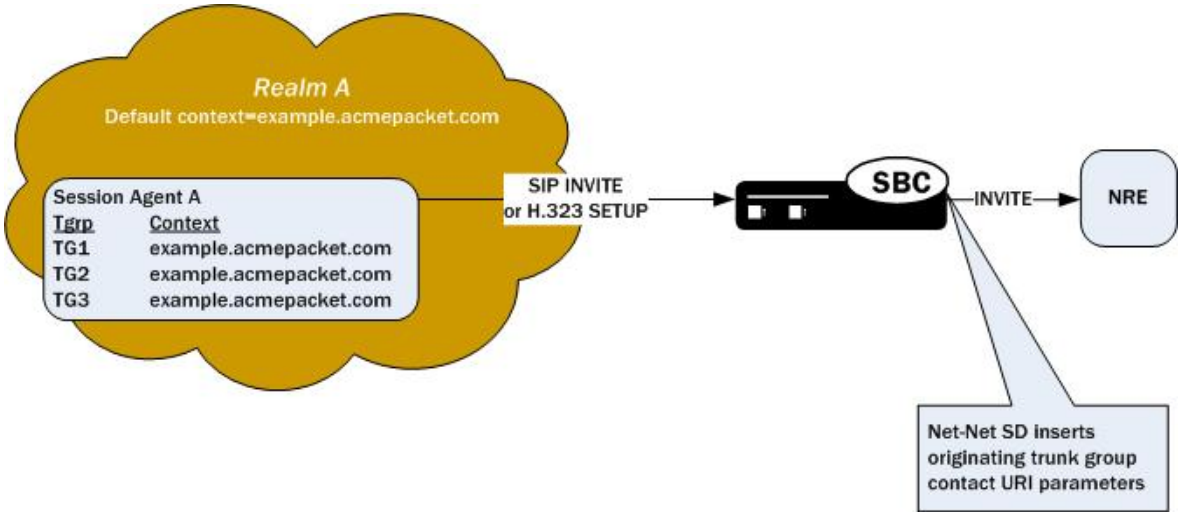
In the absence of official SIP standards for transporting trunk groups between signaling elements, the Oracle Enterprise Session Border Controller allows you to define URI parameters for use with originating and terminating trunk group URIs.

Originating Trunk Group URI Parameters and Formats

You can configure session agents and session agents groups on the Oracle Enterprise Session Border Controller to insert trunk group URI parameters in the SIP contact header. When SIP gateways comply with the IPTEL draft, they include the originating URI parameter in the SIP contact header. For those SIP and H.323 gateways that are not compliant, the Oracle Enterprise Session Border Controller inserts SIP trunk group URI parameters on the gateway's behalf.

When there are no applicable session agent or session agent group configurations, the Oracle Enterprise Session Border Controller uses the source IP address of the endpoint or gateway as the trunk group name (tgrp) parameter in the originating trunk group URI.

The following diagram shows a scenario where the Oracle Enterprise Session Border Controller inserts originating trunk group URI parameters.



There are two available formats for the originating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (identifier of the specific trunk group) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

   - tgrp=”trunk group name”
   - trunk-context=”network domain”

   The URI BNF for would appear as it does in the example directly below, where the tgrp is tg55 and the trunk-context is trunk-context is telco.example.com:

   ```
   tel:+15555551212;tgrp=tg55;trunk-context=telco.example.com
   ```

2. The second format is customized specifically for access URIs and contains two provisioned parameters: tgrp (or tgname) and context (or provstring). This appears as tgrp.context, where these definitions apply:

   - tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
   - context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

   This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

   ```
   custsite2NY-00020.type2.voip.carrier.net
   ```

   where the tgrp is custsite2NY-00020, and the context is type2.voip.carrier.net.

   The BNF for an access URI conforms to the following:

   SIP-URI = "sip:" [userinfo] hostport uri-parameters [headers]

   uri-parameters = *(";" uri-parameter)

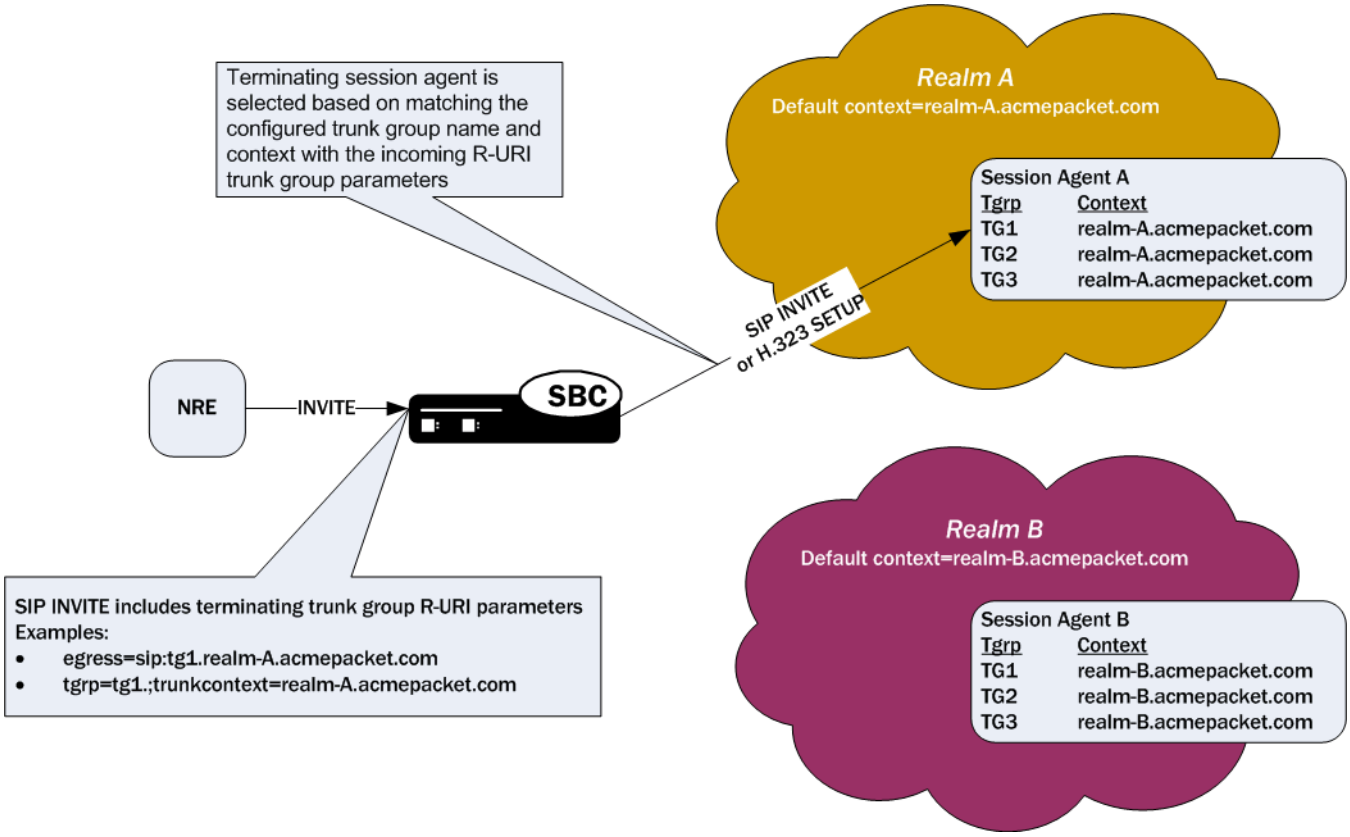
   uri-parameter = transport-param / user-param / method-param

   / ttl-param / maddr-param / lr-param / other-param

   ```
   other-param = accessid / pname [ '=' pvalue ]
   accessid = "access=" accessURI
   accessURI = scheme tgname ["." provstring]
   scheme = "sip:" / token
   tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
       alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
   provstring = *(domain ".") toplabel # up to 24 characters
   toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
   domain = alphanum/ alphanum *( alphanum / "-" ) alphanum
   ```

Terminating Trunk Group URI Parameters and Formats

Terminating trunk group URI parameters appear in the R-URI, and they can be included in by a network routing element to instruct the Oracle Enterprise Session Border Controller which egress trunk groups to use. By matching the trunk group URI parameter with configured session agents or session agent groups, the Oracle Enterprise Session Border Controller can locate the terminating gateway. The trunk group name can also be expressed as the IP address of the terminating gateway.



In the absence of official SIP standards for transporting trunk groups between signaling elements, the Oracle Enterprise Session Border Controller allows you to define the URI parameters used in terminating trunk groups.

There are two available formats for the terminating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (which can be either a trunk group name or an IP address) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

   - tgrp=”trunk group name”
   - trunk-context=”network domain”

   An example R-URI with terminating trunk group parameters appears as follows, where the tgrp is TG2-1 and the context is isp.example.net@egwy.isp.example.net:

   ```
   INVITE sip:+15555551212;tgrp=TG2-1;trunk-
   context=isp.example.net@egwy.isp.example.net SIP/2.0
   ```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: tgrp (or tgname) and context (or tgdomain). This appears as tgrp.context (or tgname.tgdomain), where definitions apply:

   - tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
   - context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```
SIP-URI = "sip:" [userinfo ] hostport uri-parameters [headers ]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
            / ttl-param / maddr-param / lr-param / other-param
other-param = egressid  / pname [ '=' pvalue ]
egressid = "egress=" egressURI
egressURI = scheme tgname ["." tgdomain]
scheme = "sip:" / token
tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
    alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
tgdomain = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
domain = alphanum/ alphanum *( alphanum / "-" ) alphanum
```

For all trunk group URI support, you must set the appropriate parameters in the SIP manipulations configuration and in the session agent or session agent group configurations.

In the originating trunk group URI scenario, a call arrives at the Oracle Enterprise Session Border Controller from a configured session agent or session agent group. If this session agent or session agent group has the appropriate trunk group URI parameters and inbound manipulation rules configured, the Oracle Enterprise Session Border Controller then looks to the SIP manipulations configuration and add the trunk group URI information according to those rules. Those rules tell the Oracle Enterprise Session Border Controller where and how to insert the trunk group URI information, and the system forwards the call.

In the terminating trunk group scenario, a call arrives at the Oracle Enterprise Session Border Controller from, for instance, a call agent. This call contains information about what trunk group to use. If the information matches a session agent or session agent group that has outbound manipulation rules configured, the Oracle Enterprise Session Border Controller will then look up the SIP manipulations configuration and strip information according to those rules. Those rules tell the Oracle Enterprise Session Border Controller where and how to remove the information, and the Oracle Enterprise Session Border Controller forwards the call.

SIP Header and Parameter Manipulation

SIP header and parameter manipulation is its own configuration where you can set up rules for the addition, removal, and modification of a SIP header or the elements of a SIP header. For example, you can set up the configuration to add a URI parameter to the URI in a SIP header or replace an FQDN with in IP address. For trunk group URI support, this configuration tells the Oracle Enterprise Session Border Controller where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

These manipulations can be applied at the realm or at the session agent level.

Trunk Group Routing

You can configure SIP interfaces (using the ACLI term-tgrp-mode parameter) to perform routing based on the trunk group information received in SIP requests. There are three options: none, IPTEL, and egress URI.

- If you leave this parameter set to none (its default), the Oracle Enterprise Session Border Controller will not look for or route based on terminating trunk group URI parameters
- When you set this parameter to either iptel or egress-uri and the incoming request has the trunk group parameter of this type (IPTEL or egress URI), the Oracle Enterprise Session Border Controller will select the egress next hop by matching the “tgrp” and trunk context with a configured session agent or session agent group.

  If the received terminating trunk group URI parameters include an IP address, the egress next hop is the IP address specified. The Oracle Enterprise Session Border Controller determines the egress realm by matching the trunk context it receives with the trunk context you configure for the realm.
- If the incoming request does not have trunk group parameters or it does not have trunk group parameters of the type that you configure, the Oracle Enterprise Session Border Controller uses provisioned procedures and/or local policy for egress call routing.

The Oracle Enterprise Session Border Controller returns errors in these cases:

- If the terminating trunk group URI parameters do not identify a local Oracle Enterprise Session Border Controller session agent or session agent group, then the Oracle Enterprise Session Border Controller returns a SIP final response of 488 Not Acceptable Here.
- If the Oracle Enterprise Session Border Controller receives a SIP INVITE with terminating trunk group URI parameters that do not match the specified syntax, the Oracle Enterprise Session Border Controller returns a 400 final response with the reason phrase Bad Egress=Parameters.

Trunk Group URIs and SIP Registration Caching

For calls where SIP registration caching is used, you will need to set certain parameters that enable the Oracle Enterprise Session Border Controller to preserve trunk group URI parameters on the outgoing side.

- For SIP-SIP calls, you set the preserve-user-info option in the SIP interface configuration.
- For SIP-H.323 calls requiring IWF, you set the preserve-user-info-sa option in the session agent configuration.

Trunk Group URI Configuration

Before you configure your Oracle Enterprise Session Border Controller to support trunk group URIs, you need to determine:

- How you want to manipulate SIP headers (entered in the SIP header manipulations configuration)
- For terminating trunk group routing, the trunk group mode you want to use (none, IPTEL, or egress URI); this decides routing based on trunk group information
- The trunk group name and context to use entered in a session agent or session agent group configuration
- Whether you are using originating or terminating trunk group URIs (entered in the session agent configuration)
- The trunk group context for use in a realm configuration, in case the trunk group name in the session agent or session agent group does not have a context

Configuring SIP Manipulations

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The new-value parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are $TRUNK_GROUP and $TRUNK_GROUP_CONTEXT

Setting the Trunk Group URI Mode for Routing

To set the mode for routing for terminating trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. term-tgrp-mode—Set the mode that you want to use for routing for terminating trunk group URIs. The default is none. Your choices are:
   - none—Disables routing based on trunk groups
   - iptel—Uses trunk group URI routing based on the IPTEL formats
   - egress-uri—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. out-manipulationid—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the Oracle Enterprise Session Border Controller via this session agent. There is no default.
5. in-manipulationid—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic entering the Oracle Enterprise Session Border Controller via this session agent. There is no default.
6. trunk-group—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Oracle Enterprise Session Border Controller will use the one you set in the realm for this session agent.

   Your ACLI entries for this list must be one of these formats: tgrp:context or tgrp.context.

   To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

   ```
   ACMEPACKET(session-agent)# trunk-group (tgrp1:context1 tgrp2:context2)
   ```

7. options—If you want to configure trunk group URIs for SIP-H.323 calls that use the IWF and you are using SIP registration caching, you might need to add the preserve-user-info-sa to your list of session agent options.

   If you are adding this option to a new session agent, you can just type options, a Space, and preserve-user-info-sa.

   If are adding this to an existing session agent, you must type a plus (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +preserve-user-info-sa.

Configuring a Session Agent Group for Trunk Group URIs

In a session agent group, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent group for trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-group and press Enter.

   ```
   ACMEPACKET(session-router)# session-group
   ACMEPACKET(session-agent-group)#
   ```

4. trunk-group—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Oracle Enterprise Session Border Controller will use the one you set in the realm for this session agent group.

   Your ACLI entries for this list must take one of these formats: tgrp:context or tgrp.context.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent-group)# trunk-group (tgrp1:context1 tgrp2:context2)
```

Setting a Trunk Group Context in a Realm

You can set trunk group contexts at the realm level, which will be used by all session agents and session agent groups if there is no context specified in their configurations.

The realm trunk group URI context accommodates the IPTEL and the custom format.

To configure a trunk group context for a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. trunk-context—Enter the trunk group context to use for this realm. There is no default.

Using this Feature with a SIP Interface

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the preserve-user-info option for that SIP interface.

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-group and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. options—Add support for trunk group URIs with SIP interface that uses registration caching.

   If you are adding this option to a new SIP interface, you can just type options, a Space, and preserve-user-info.

   If are adding this to an existing SIP interface, you must type a plus (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +preserve-user-info.

Example 1 Adding Originating Trunk Group Parameters in IPTEL Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTEL format.

```
sip-manipulation
        name                            add_iptel
        header-rule
                name                            contact
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            tgrp
                        type                            uri-user-param
                        action                          add
                        match-val-type                  any
```

```
                        match-value
                        new-value                       $TRUNK_GROUP
element-rule
                        name                            trunk-context
                        type                            uri-user-param
                        action                          add
                        match-val-type                  any
                        match-value
                        new-value                       $TRUNK_GROUP_CONTEXT
```

Example 2 Adding Originating Trunk Group Parameters in Custom Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in custom format.

```
sip-manipulation
        name                            add_att
        header-rule
                name                            contact
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            egressURI
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        match-value
                        new-value                       "sip:"+$TRUNK_GROUP
+"."+$TRUNK_GROUP_CONTEXT
```

Example 3 Removing IPTEL Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove IPTEL trunk groups names.

```
sip-manipulation
        name                            strip_iptel
        header-rule
                name                            request-uri
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            tgrp
                        type                            uri-user-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
element-rule
                        name                            trunk-context
                        type                            uri-user-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
```

Example 4 Removing Custom Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove custom trunk groups names.

```
sip-manipulation
        name                            strip_egress
```

```
        header-rule
                name                                    request-uri
                action                                  manipulate
                match-value
                msg-type                                any
                element-rule
                        name                                    egressURI
                        type                                    uri-param
                        action                                  delete-element
                        match-val-type                          any
                        match-value
                        new-value
```

Emergency Session Handling

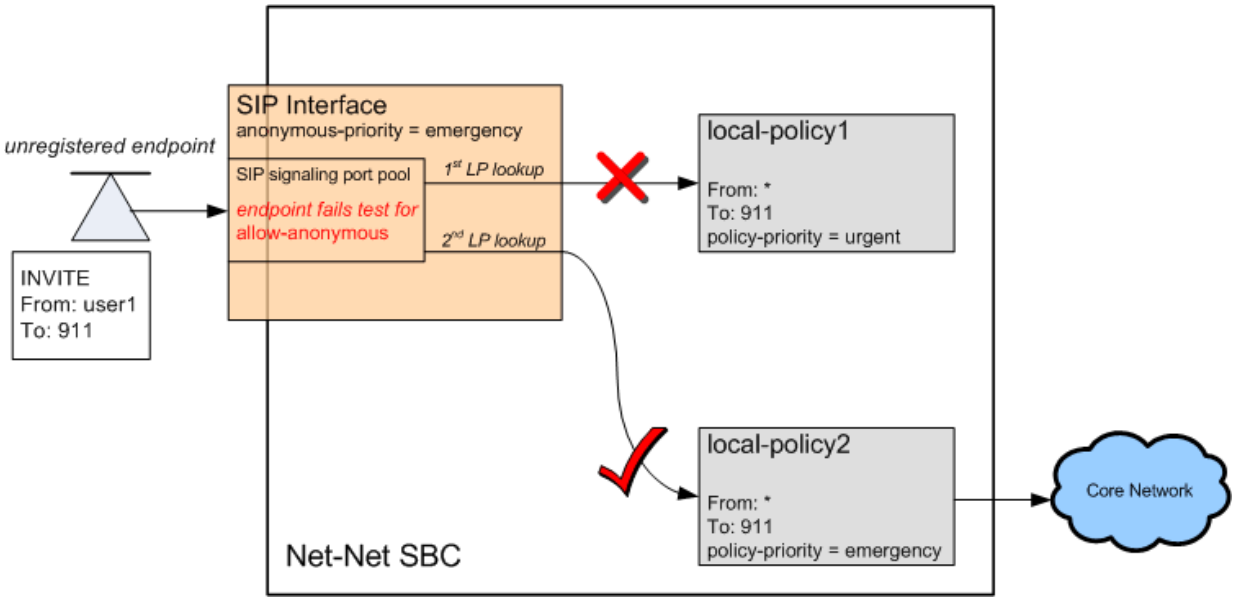
The Oracle Enterprise Session Border Controller provides a mechanism to handle emergency sessions from non-allowed endpoints. An endpoint is designated as non-allowed if it fails the admission control criteria specified by the allow-anonymous parameter in the SIP Ports configuration element.

When the Oracle Enterprise Session Border Controller receives a non-allowed emergency request, it performs a local policy lookup for a matching local policy. An emergency local policy could be configured to match if the To: header in a SIP message was addressed to 911.

An emergency policy priority selection criteria has been added to both the SIP interface and the local policy configuration elements. In the SIP interface, the parameter is called anonymous-priority. In the local policy, the parameter is called policy-priority.

For the Oracle Enterprise Session Border Controller to choose a local policy to route an emergency call, the emergency policy priority value on the local policy must be equal to or greater than the emergency policy priority value on the SIP interface where the emergency message was received. In this scheme, an emergency policy priority value of none is the lowest value and an emergency policy priority value of emergency is the highest.

When a match is made between all existing local policy criteria and the emergency policy priority, the emergency call will be sent to the core network according to the chosen local policy. In addition, the policy priority value of the chosen local policy is inserted into the Priority header of the core-bound SIP message..



Emergency Session Handling Configuration Procedures

Note the value of the allow-anonymous parameter in the SIP interface's SIP Ports for the incoming interface you are configuring. When an incoming emergency call from an unregistered endpoint can not be characterized by this setting, the Oracle Enterprise Session Border Controllerwill use the following means to route the call.

Set the anonymous-priority parameter in the incoming SIP interface. This parameter specifies that for an INVITE received from an anonymous endpoint, the Oracle Enterprise Session Border Controller will choose a local policy of equal or greater policy priority for outbound routing.

Next, set the policy-priority parameter located in the local-policy configuration element. Most likely, this local policy will route messages to SIP devices that act on emergency calls. The local policy is selected when its value (or above) matches the anonymous-priority parameter in the sip-interface that receives the incoming phone call from an unregistered endpoint.

The enumerated values for both the anonymous-priority and policy-priority are: none, normal, non-urgent, urgent, emergency.

Emergency Session Handling Configuration

To set the anonymous priority for a message received in a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. Type select and the number of the SIP interface you want to configure.

   ```
   ACMEPACKET(sip-interface)# select 1
   ```

5. anonymous-priority—Set the policy priority for this SIP interface. It is used to facilitate emergency sessions from unregistered endpoints. This value is compared against the policy-priority parameter in the local-policy configuration element. The default is none. The valid values are:

   - none | normal | non-urgent | urgent | emergency

     This completes the configuration.

     ```
     ACMEPACKET(sip-interface)# anonymous-priority emergency
     ```

6. Save your work using the ACLI done command.

Setting Policy Priority

To set the policy priority for a local policy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type local-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# local-policy
   ACMEPACKET(local-policy)#
   ```

4. Type select and the number of the local policy you want to configure.

```
ACMEPACKET(local-policy)# select 1
```

5. policy-priority—Enter the policy priority for this local policy. It is used to facilitate emergency sessions from unregistered endpoints. This value is compared against the anonymous-priority parameter in the sip-interface configuration element. The default is none. The valid values are:

   - none | normal | non-urgent | urgent | emergency

   This completes the configuration.

```
ACMEPACKET(local-policy)# anonymous-priority emergency
```

6. Save your work using the ACLI done command.

Fraud Prevention

The Oracle Enterprise Session Border Controller can constrain outgoing SIP messages to a maximum size in bytes in order to support fraud prevention techniques. If a message does exceed the configured size, it is dropped. A SIP message can be constrained from 0 to 65535 bytes, with a default value of 4096 bytes.

Fraud Prevention Configuration

To set a maximum SIP message size:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. Type select to configure the existing sip config.

```
ACMEPACKET(sip-config)# select
```

5. sip-message-len—Set the size constraint in bytes of a SIP message. The default is 4096. The valid range is:

   - Minimum—0
   - Maximum—65535

   This completes the configuration.

```
ACMEPACKET(sip-config)# sip-message-len 5000
```

6. Save your work using the ACLI done command.

SIP Early Media Suppression

This section explains how to configure SIP early media suppression, which lets you determine who can send early media and in what direction. Early media are the RTP/RTCP packets sent from the called party to the caller, or vice versa, before a session is fully established (before a 200 OK is received). When the Oracle Enterprise Session Border Controller receives an INVITE message with SDP, it can forward media packets to the calling endpoint as soon as it forwards the INVITE to the next hop. It can also forward media packets received from the calling endpoint to the called endpoint as soon as the Oracle Enterprise Session Border Controller receives SDP in a SIP response to the INVITE, usually a provisional message. This allows for any early media to be played, such as remote ringback or announcement.

Early media can be unidirectional or bidirectional, and can be generated by the caller, the callee, or both.

With early media suppression, you can block early media until the call is established. You can define which outbound realms or next hop session agents are allowed to send or receive early media. early media suppression only applies to RTP packets. RTCP packets received by Oracle Enterprise Session Border Controller are still forwarded to their destination in both directions, unless an endpoint is behind a NAT and the media manager has not been enabled for RTCP forwarding.

Note: To use early media suppression, you cannot configure media release of any kind: same-realm, same-network, or multiple-system media release.

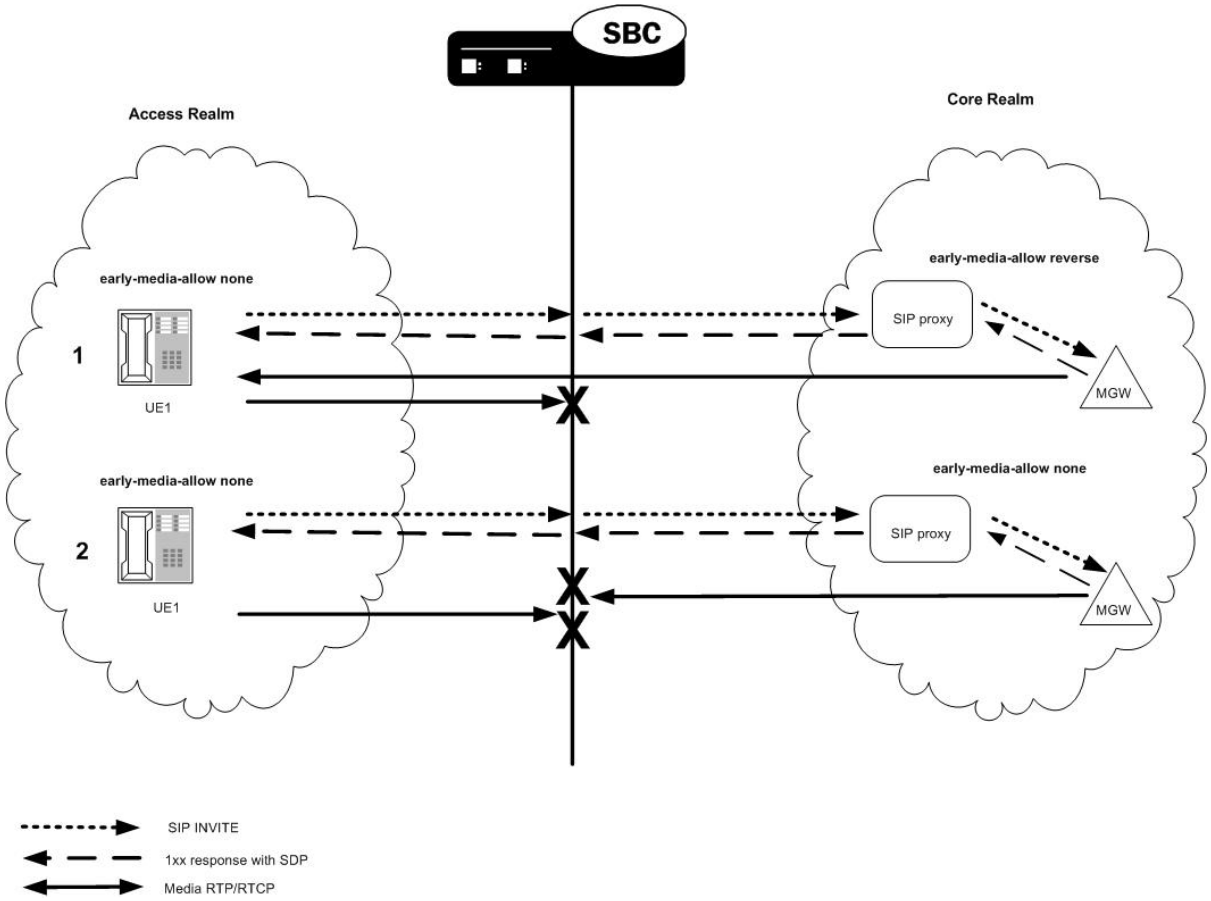
With the SIP-based addressing, early media suppression is based on the outbound SIP interface realms and the value of their early-media-allow parameter. When the Oracle Enterprise Session Border Controller forwards a SIP Invite out a SIP interface, the outbound realm is chosen based on the SIP layer information, such as the session agent for the next-hop or the address prefix of the next-hop SIP device. The matching realm's early-media-allow parameter value then applies to either allow all, block all, or block one-way early media until a 200 ok is received. At that point bidirectional media is allowed. The decision is based on SIP-layer addressing of next-hops.

You configure a rule for a realm or a session agent to use early media suppression. An early media suppression rule specifies whether you want to prevent early media in any direction, allow early media going to the calling endpoint in the reverse direction, or allow early media in both directions. The forward direction is when the packets flow from the caller to the called party. The reverse direction is when the packets flow from the called party to the caller.

The early media suppression rule is applied to a session. When the Oracle Enterprise Session Border Controller initiates a new session, it first checks whether the next hop is a session agent and if so, whether an early media suppression rule has been configured it. If an early media suppression rule is found, the Oracle Enterprise Session Border Controller enforces it. If the next hop is not a session agent or no early media suppression rule is configured, the Oracle Enterprise Session Border Controller checks whether an early media suppression rule has been configured for the outbound realm. If it finds one, it enforces it.

Example

The following illustration shows two examples of early media suppression.



1. Caller UE1 makes a call to the PSTN media gateway (MGW). The INVITE traverses from UE1 to the Oracle Enterprise Session Border Controller through the softswitch to the MGW. The Oracle Enterprise Session Border Controller allows early media from the core to reach UE1.
2. The PSTN MGW makes a call to UE1. The INVITE traverses to the Oracle Enterprise Session Border Controller and to UE1. The Oracle Enterprise Session Border Controller blocks all early media to and from UE1 until a 200 OK is received.

Early Media Suppression Support

The Oracle Enterprise Session Border Controller supports suppressing early media in the following directions no matter which side makes the SDP offer, until it receives 200 OK for an INVITE:

- Forward direction based on the outbound realm or next-hop session agent
- Forward and reverse directions based on the outbound realm or next-hop session agent.

The Oracle Enterprise Session Border Controller allows all media when a 200 OK response is received for the INVITE, regardless of whether the 200 OK response contains SDP.

Call Signaling

The Oracle Enterprise Session Border Controller media manager performs early media suppression according to an early media suppression rule. No change has been made to call signaling. For SIP, the Oracle Enterprise Session Border Controller still forwards SDP received in an INVITE request or response after performing a NAT to the media connection address. After which, the Oracle Enterprise Session Border Controller is ready to receive media packets from the endpoints. If an early media suppression rule has been configured, the Oracle Enterprise Session Border Controller drops the packets going in the direction being specified by the rule.

For a H.323 to SIP call, early media suppression rule does not change how the Oracle Enterprise Session Border Controller performs H.225/Q.931 call signaling and starts the H.245 procedure (if required) to establish logical channels for early media on the H.323 leg of the call.

Suppression Duration

When early media suppression is enabled in a session, the block lasts until the session is established. For a SIP to SIP call or an H.323 to SIP call, a session is established when the system receives a 200 OK response to the INVITE. A 200 OK response to the INVITE terminates early media suppression, even when it does not contain a SDP. (A 200 OK response to a PRACK or an UPDATE request does not terminate early media suppression.) After a session is established, the Oracle Enterprise Session Border Controller can receive a change in media session (for example, a re-INVITE with a new SDP) without an early media suppression rule blocking the media.

About the Early Media Suppression Rule

An early media suppression rule is configured in the form of a permission. It specifies whether early media is allowed in both directions, the reverse direction only or not at all. Reverse direction media is media sent in the upstream direction towards the calling endpoint.

Session Agent Rule

The next-hop session agent's early media suppression rule is applied regardless of whether the media packet's source or destination address is the same as the session agent's address. For example, if the session's next hop session agent is 10.10.10.5 but the SDP in a 183 response specifies 10.10.10.6 as its connection address.

Rule Resolution

When the call's next hop is a session agent and both the outbound realm of the call and the session agent have an early media suppression rule, the session agent's early media suppression rule takes precedence. If the session agent's early media suppression rule has not been configured, the outbound realm's early media suppression rule is used, if configured.

Selective Early Media Suppression

Normally, the Oracle Enterprise Session Border Controller performs early media blocking based on destination realm. Calls to such realms are prohibited from sending and receiving RTP until a SIP 200 OK response is received, and you can set the direction of the blocked media.

While decisions to block early media are customarily based on SIP-layer addressing, there are cases when the Oracle Enterprise Session Border Controller can reject early media based on the SDP address in the SDP answer for a 1XX or 2XX response. By comparing the SDP address with the realm prefix or additional prefix address, it can block early media for matching realms. For these cases, you define global or signaling realms—ones that are not tied to SIP interfaces, but which establish additional address prefixes and rules for blocking early media.

This way, the Oracle Enterprise Session Border Controller blocks all early media for SIP interface realms, but can accept it for global realms that reference media or PSTN gateways. This configuration allows early media for calls destined for the PSTN, and blocks it for user-to-user and PSTN-to-user calls.

Selective early media suppression addresses the fact that some service providers need to allow early media for certain user-to-user and PSTN-to-user calls to support, for example, custom ringback tones. The enhancements also address the fact that Oracle Enterprise Session Border Controllers can themselves lose the ability to decide whether or not early media should be blocked when confronted with hairpinned call flows, or with traffic that traverses multiple Oracle Enterprise Session Border Controllers.

To address this need, you can configure realm groups. Realm groups are sets of source and destination realms that allow early media to flow in the direction you configure. For example, you can set up realm groups to allow media from PSTN realms to user realms so that users can listen to PSTN announcements, but prohibit early media from user realms to PSTN realms.

Note: The source and destination realms you add to your lists need to be a global signaling realm matching the caller's SDP address prefix or a SIP realm.

Configuring the Realm

To configure the realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```

4. If configuring an existing realm, enter the select command to select the realm.
5. early-media-allow—Enter the early media suppression rule for the realm. The valid values are:

   - none—No early media is allowed in either direction
   - both—Early media is allowed in both directions
   - reverse—Early media received by Oracle Enterprise Session Border Controller in the reverse direction is allowed

   There is no default value. If you leave this parameter blank, early media is allowed in either direction. You can use the following command to clear this parameter:

```
early-media-allow ()
```

6. Save and activate your configuration.

   For example:

```
realm-config
        identifier                      access1
        addr-prefix                     192.168.1.0/24
        network-interfaces
                                        media:0
        mm-in-realm                     enabled
        mm-in-network                   enabled
        msm-release                     disabled
        qos-enable                      disabled
        max-bandwidth                   0
        max-latency                     0
        max-jitter                      0
        max-packet-loss                 0
        observ-window-size              0
        parent-realm
        dns-realm
        media-policy
        in-translationid
        out-translationid
        class-profile
        average-rate-limit              0
        access-control-trust-level      none
        invalid-signal-threshold        0
        maximum-signal-threshold        0
        deny-period                     30
        early-media-allow               none
        last-modified-date              2006-02-06 13:09:20
```

Configuring Session Agents

If you do not configure early media suppression for a session agent, the early media suppression for the outbound realm is used, if configured.

To configure session agents:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. If configuring an existing session agent, enter the select command to select the session agent.
5. early-media-allow—Enter the early media suppression rule for the session agent. The valid values are:

   - none—No early media is allowed in either direction
   - both—Early media is allowed in both directions
   - reverse—Early media received by Oracle Enterprise Session Border Controller in the reverse direction is allowed

   There is no default value. If you leave this parameter blank, early media is allowed in either direction. You can use the following command to clear this parameter:

   ```
   early-media-allow ()
   ```

6. Save and activate your configuration.

   For example:

```
session-agent
        hostname                        cust1
        ip-address                      192.168.1.24
        port                            5060
        state                           enabled
        app-protocol                    SIP
        app-type
        transport-method                UDP
        realm-id                        access1
        description
        carriers
        allow-next-hop-lp               enabled
        constraints                     disabled
        max-sessions                    0
        max-outbound-sessions           0
        max-burst-rate                  0
        max-sustain-rate                0
        time-to-resume                  0
        ttr-no-response                 0
        in-service-period               0
        burst-rate-window               0
        sustain-rate-window             0
        req-uri-carrier-mode            None
        proxy-mode
        redirect-action
        loose-routing                   enabled
        send-media-session              enabled
        response-map
        ping-method
        ping-interval                   0
        media-profiles
        in-translationid
        out-translationid
        trust-me                        disabled
```

```
        early-media-allow               reverse
        last-modified-date              2006-05-06 13:26:34
```

Configuring Realm Groups

To configure a realm group for selective early media suppression:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type realm-group and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-group
   ACMEPACKET(realm-group)#
   ```

4. name—Enter the name of the realm group.
5. source-realm—Enter the list of one or more global/SIP realms that you want to designate as source realms for the purpose of blocking early media; this is the realm identifier value for the realms you want on the list. Values in this list refer to calling SDP realms; this parameter has no default. To enter more than one realm in the list, list all items separated by a comma and enclose the entire entry in quotation marks:

   ```
   ACMEPACKET(realm-group)# source-realm Private, Public
   ```

   To add a realm to the list, use the plus sign (+) in front of each new entry.

   ```
   ACMEPACKET(realm-group)# source-realm +Private
   ```

   You can also remove single items in the list by using the minus sign (-) directly in front of the realm identifier.

   ```
   ACMEPACKET(realm-group)# source-realm -Private
   ```

6. destination-realm—Enter the list of one or more global/SIP realms that you want to designate as destination realms for the purpose of blocking early media; this is the realm identifier value for the realms you want on the list. Values in this list refer to called SDP realms; this parameter has no default. To enter more than one realm in the list, list all items separated by a comma and enclose the entire entry in quotation marks:
7. ACMEPACKET(realm-group)# source-realm Private, Public

   To add a realm to the list, use the plus sign (+) in front of each new entry.

   ```
   ACMEPACKET(realm-group)# destination-realm +Private
   ```

   You can also remove single items in the list by using the minus sign (-) directly in front of the realm identifier.

   ```
   ACMEPACKET(realm-group)# destination-realm -Private
   ```

8. early-media-allow-direction—Set the direction for which early media is allowed for this realm group. Valid values are:

   - none—Turns off the feature for this realm group by blocking early media
   - reverse—Allows early media to flow from called to caller
   - both (default)—Allows early media to flow to/from called and caller

9. Save and activate your configuration.

SDP-Response Early Media Suppression

This section explains how to configure SDP-response early media suppression, which can be used when the Oracle Enterprise Session Border Controller is deployed after a softswitch or proxy in the signaling path. In this deployment, user endpoints and gateways communicate directly with the softswitch or proxy, which in turn sends call signaling to the Oracle Enterprise Session Border Controller. The call signaling gets sent back to the same or different softswitch or proxy. Because the Oracle Enterprise Session Border Controller does not communicate with the endpoints or

gateways that are the media terminators, early media suppression for this deployment must use SDP-based addressing rather than the SIP-based addressing (described in the SIP Early Media Suppression section in this technical notice).

Using this feature lets you configure specific IP addresses for which early media should not be suppressed, based on SDP addressing. The Oracle Enterprise Session Border Controller checks the SDP addresses in SIP responses against these IP address or address ranges to determine on which media gateway a call terminates.

SIP-Based Addressing

With the SIP-based addressing described in the SIP Early Media Suppression section, early media suppression is based on the outbound SIP interface realms and the value of their early-media-allow parameter. When the Oracle Enterprise Session Border Controller forwards a SIP Invite out a SIP interface, the outbound realm is chosen based on the SIP layer information, such as the session agent for the next-hop or the address prefix of the next-hop SIP device. The matching realm's early-media-allow parameter value then applies to either allow all, block all, or block one-way early media until a 200 ok is received. At that point bidirectional media is allowed. The decision is based on SIP-layer addressing of next-hops.

SDP-Based Addressing

SDP-response early media suppression follows the same sequence described for SIP-based addressing with one exception. A provisional response with SDP media can make the Oracle Enterprise Session Border Controller select a new early-media-allow rule from another realm, based on the addressing inside the responding SDP.

When the SDP-response early media suppression feature is enabled, the Oracle Enterprise Session Border Controller searches the outbound SIP interface's realms for a matching address prefix with the connection address in the responding SDP. If it finds a match, it uses the early-media-allow parameter value of that realm until the 200 OK message is received, then bidirectional media is allowed regardless. If the Oracle Enterprise Session Border Controller does not find a match, it searches all of the global realms for one. If it finds a match, the Oracle Enterprise Session Border Controller uses that realm's early-media-allow parameter value. If it does not find a match in the global realm(s), the Oracle Enterprise Session Border Controller continues to use the previous early-media-allow parameter value.

Global Realms

Global realms are realms that are not parents or children of any other realms, do not have defined SIP interfaces and ports (or any signaling interface or stack), and are configured to use the network interface lo0:0. They are special realms, applicable system-wide, and are currently only used for this feature. The only global realm configuration parameters applicable to early media suppression are:

- addr-prefix
- additional-prefixes
- early-media-allow
- network-interface (which must be set to lo0:0)

Additional Prefixes

You can specific additional prefixes in addition to that of the addr-prefix parameter you configure for a realm. For example, you can configure a global realm with additional address prefixes to specify the IP addresses (or ranges of addresses) of the media gateways that are allowed to send and receive early media. This overrides the SIP interface realm's early media blocking settings.

You can also enter additional prefixes in non-global realms. These additional prefixes function the same as would multiple values in the addr-prefix parameter (which only takes one value), except addresses in additional-prefixes are not used for SIP NATs.

Using the SDP-Response Early Media Suppression Rule

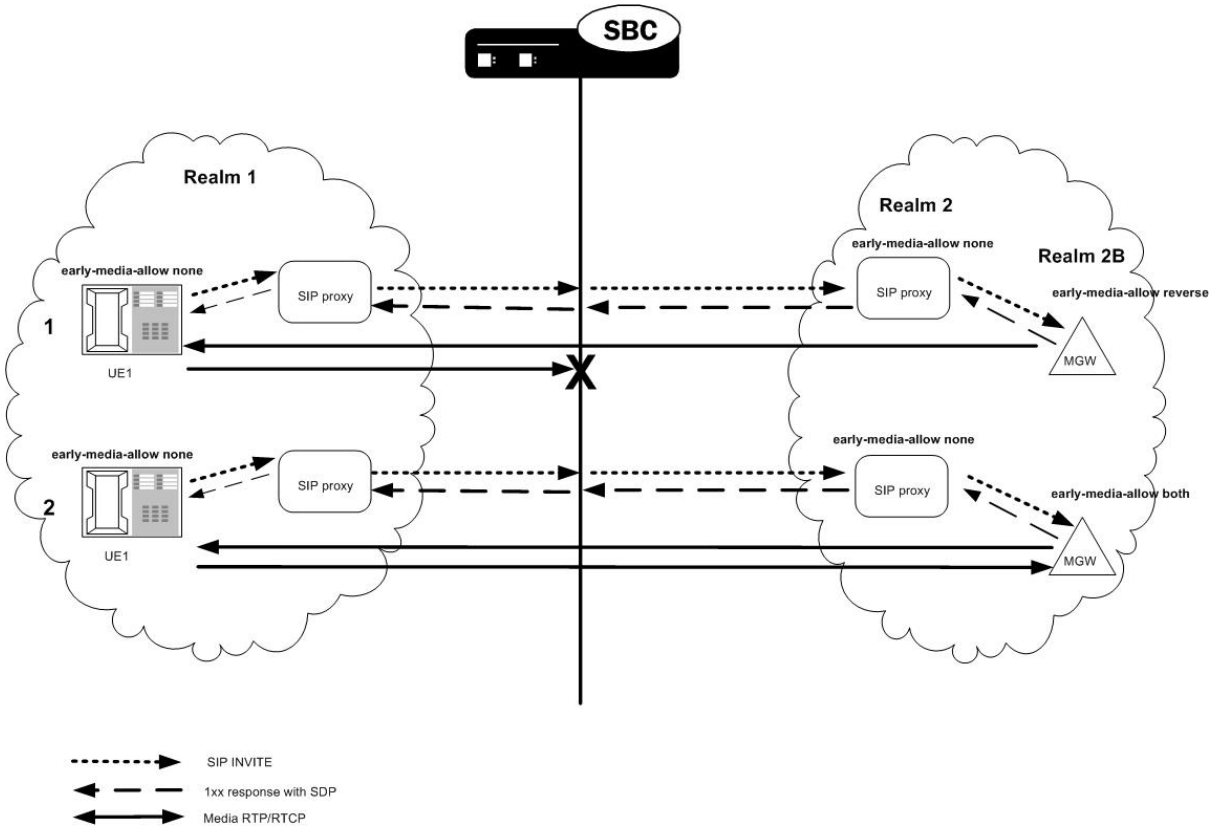
To use SDP-response early media suppression, you must add the early-media-sdp-realms option to the SIP interface configuration that interfaces with the next-hop device, such as the supported softswitch.

When the Oracle Enterprise Session Border Controller receives a provisional response that includes SDP from the called endpoint, and the early-media-sdp-realms option is active in the outgoing SIP interface of the call, it first searches the realms that apply to the outgoing SIP interface. If it does not find a realm, the Oracle Enterprise Session Border Controller searches the global realms. If the search yields a new realm that is not the SIP interface realm, its early media suppression rule (if any) replaces the existing one. Only the early media suppression rule of the new realm is applied to the call. Other realm properties from the outbound realm remain applicable to the call. If no new realm is found, the early media policy of the outgoing SIP interface realm is applied.

TheOracle Enterprise Session Border Controller allows media when the SDP media connect address in a response matches one of a configured list of IP address ranges defined in a realm and the realm has early media allowed. You need to configure specific a IP address or address range to specify which media gateways should not be suppressed based on SDP media addresses. The IP addresses are checked against the SDP being received. The decision for suppression is based on whether the matching realm allows early media. The early media will be suppressed if the matching realm does not allow early media or if there is no match and the outbound SIP interface ream does not allow early media.

Example

The following illustration shows two examples of SDP-response early media suppression.



Configuring SDP-Response Early Media Suppression

To configure SDP-response early media suppression:

1. Add the early-media-sdp-realms option to the SIP interface that interfaces with the softswitch.
2. Configure the SIP interface realm with an early media suppression rule that blocks all early media.
3. Configure either or both of the following:
   - One or more of the SIP realm's child realms, each with an early media suppression rule that allows all or reverse direction early media and a list of additional prefixes that specifies the IP addresses of the media gateways, or a range of IP addresses that includes the media gateways. Early media is allowed from these gateways only for calls that signals through this SIP interface.

- One or more realms that has the network interface equal to lo0:0, an early media suppression rule that allows all or reverse direction early media and a list of additional prefixes that specifies the IP addresses of the media gateways, or a range of IP addresses that includes the media gateways. Early media is allowed from these gateways regardless of interface.

Configuring the SIP Interface

To configure a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

   From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.

4. If configuring an existing interface, enter the select command to select the interface.
5. options—Enter early-media-sdp-realms as the option. If adding to an existing list of options, use a preceding plus (+) sign.

```
options +early-media-sdp-realms
```

6. Continue to the next section to configure the outbound realm.

   For example:

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)# options + early-media-sdp-realms
ACMEPACKET(sip-interface)# done
sip-interface
        state                           enabled
        realm-id                        access1
        sip-port
                address                         192.168.1.30
                port                            5060
                transport-protocol              UDP
                allow-anonymous                 all
        carriers
        proxy-mode                      Proxy
        redirect-action
        contact-mode                    maddr
        nat-traversal                   none
        nat-interval                    30
        registration-caching            disabled
        min-reg-expire                  300
        registration-interval           3600
        route-to-registrar              disabled
        teluri-scheme                   disabled
        uri-fqdn-domain
        options                         early-media-sdp-realms
        trust-mode                      all
        last-modified-date              2006-05-10 18:27:31
```

Configuring a Realm

To configure a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. If configuring an existing realm, enter the select command to select the realm.
5. early-media-allow—Enter the early media suppression rule for the realm. The valid values are:
   - both—Early media is allowed in both directions
   - reverse—Early media received by Oracle Enterprise Session Border Controller in the reverse direction is allowed
   - none—Early media is blocked
6. additional-prefixes—Enter a single or a comma-delimited list of IP address prefixes to use in addition to the value of the addr-prefix parameter.

```
<IPv4> [/<number of bits>]
```

   <IPv4> is a valid IPv4 address and <number of bits> is the number of bits to use to match an IP address with the address prefix. Not specifying <number of bits> implies that all 32 bits are used for matching.

   Enclose the list between quotes if there is any space between a comma and the next address prefix.

   You can add and remove address prefixes to and from the list:

   - add-additional-prefixes adds one or more additional prefixes

```
add-additional-prefixes 192.168.201.69
```

   - remove-additional-prefixes removes one or more additional prefixes

```
remove-additional-prefixes 192.168.201.69
```

     If using multiple address prefixes, enter a comma-delimited list.
7. Save and activate your configuration.

   For example:

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)# additional-prefixes
192.168.200.0/24,192.168.201.68
ACMEPACKET(realm-config)# done
realm-config
        identifier                      early-media
        addr-prefix                     0.0.0.0
        network-interfaces
                                        media2:0
        mm-in-realm                     disabled
        mm-in-network                   enabled
        msm-release                     disabled
        qos-enable                      disabled
        max-bandwidth                   0
        max-latency                     0
        max-jitter                      0
        max-packet-loss                 0
        observ-window-size              0
        parent-realm
        dns-realm
        media-policy
        in-translationid
```

```
        out-translationid
        in-manipulationid
        out-manipulationid
        class-profile
        average-rate-limit              0
        access-control-trust-level      none
        invalid-signal-threshold        0
        maximum-signal-threshold        0
        untrusted-signal-threshold      0
        deny-period                     30
        symmetric-latching              disabled
        pai-strip                       disabled
        trunk-context
        early-media-allow               reverse
        additional-prefixes             192.168.200.0/24
192.168.201.69
        last-modified-date              2006-05-11 06:47:31
```

SIP Duplicate SDP Suppression

Using the strip-dup-sdp option in the SIP configuration, you can enable your Oracle Enterprise Session Border Controller to suppress a duplicate SDP answer in the reliable responses (1xx and 2xx) in an INVITE transaction.

During INVITE transactions in certain networks, SDP answers in reliable provisional responses (1xx) can cause interoperability issues. This issue occurs when the UAS includes the SDP answer in subsequent 1xx responses or in the final 2xx response, and the UAC views that inclusion as a protocol violation. The UAC does so based on the fact that the SDP answer was reliably delivered to it in a 1xx response, and so it views additional SDP information as unnecessary in and ensuing reliable 1xx or 200 OK responses.

RFCs 3216 and 3262 do not specifically call out the UAS's including SDP information in this way as a protocol violation. Still, the system allows you set enable the strip-dup-sdp option as a means of preventing the UAC from terminating sessions. With this option enabled, the Oracle Enterprise Session Border Controller removes the SDP answer in subsequent reliable provisional or final 200 OK responses if it is identical to the SDP answer previously received.

SIP Duplicate SDP Suppression Configuration

To enable duplicate SDP suppression for SIP sessions:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-config and press Enter.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI select command) the configuration that you want to edit.

4. options—Set the options parameter by typing options, a Space, and then the option name strip-dup-sdp. Then press Enter.

   ```
   ACMEPACKET(sip-config)# options +strip-dup-sdp
   ```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

SIP SDP Address Correlation

SIP SDP address correlation ensures that when the Oracle Enterprise Session Border Controller receives a request containing SDP, the L3 source address of the request is compared against the address in the c-line of the SDP. When the addresses match, the session proceeds as it normally would. If there is a mismatch, the call is rejected with the default 488 status code. You can also configure the code you want to use instead of 488.

This functionality works only with non-HNT users. The value c=0.0.0.0 is an exception and is always processed.

SIP SDP Address Correlation Configuration Address Checking

The sdp-address-check, in the enforcement-profile element can be set to enable the SDP address correlation.

To enable SDP address checking:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type enforcement-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```

4. Use the ACLI select command so that you can work with the enforcement profile configuration to which you want to add this parameter.

```
ACMEPACKET(enforcement-profile) select
```

5. sdp-address-check—Enable or disable SDP address checking on the Oracle Enterprise Session Border Controller. The default for this parameter is disabled.

```
ACMEPACKET(enforcement-profile)# sdp-address-check enabled
```

6. Save and activate your configuration.

   If a mismatch occurs and you want to reject the call with a status code other than 488, you set the code you want to use in the local response code map entries.

SIP SDP Address Correlation Configuration Mismatch Status Code

To apply a new status code to a SDP address correlation mismatch:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type local-response-map and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# local-response-map
ACMEPACKET(local-response-map)#
```

4. Type entries and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(local-response-map)# entries
ACMEPACKET(local-response-map-entry)#
```

From here, you can view the entire menu for the local response map entries configuration by typing a ?.

5. local-error—Enter sdp-address-mismatch for which to apply the new status code.
6. sip-status—Set the SIP response code to use.
7. sip-reason—Set the SIP reason string you want to use for this mapping.

```
ACMEPACKET(local-response-map-entry)# local-error sdp-addressmismatch
ACMEPACKET(local-response-map-entry)# sip-status 403
ACMEPACKET(local-response-map-entry)# sip-reason sdp address mismatch
```

8. Save and activate your configuration.

SIP SDP Address Correlation Configuration Enforcement Profile

To apply an enforcement profile to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. enforcement-profile—Enter the name of the enforcement profile you want to apply to this realm.

```
ACMEPACKET(realm-config)# enforcement-profile profile1
```

5. Save and activate your configuration.

SDP Insertion for (Re)INVITEs

If your network contains some SIP endpoints that do not send SDP in ReINVITEs but also contains others that refuse INVITEs without SDP, this feature can facilitate communication between the two types. The Oracle Enterprise Session Border Controller can insert SDP into outgoing INVITE messages when the corresponding, incoming INVITE does not contain SDP.

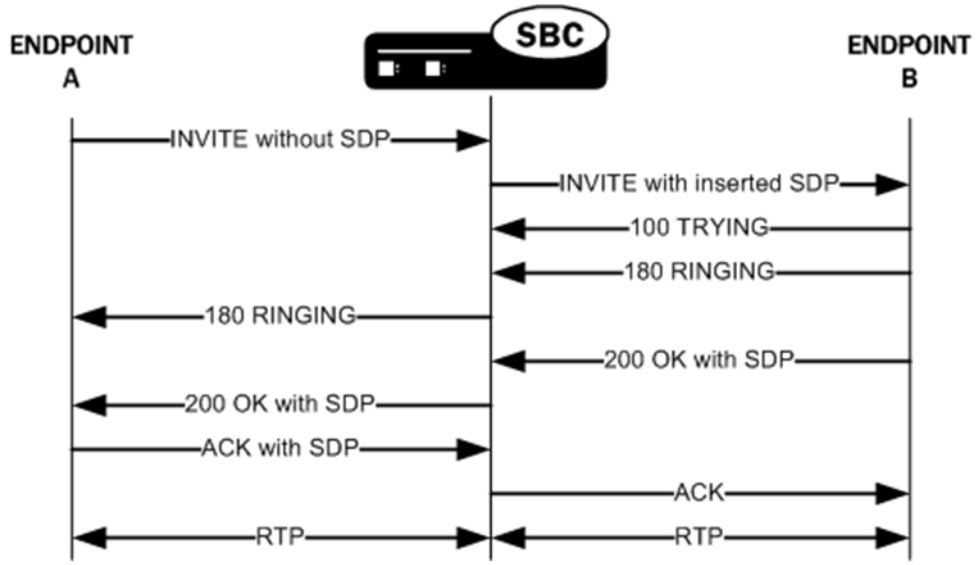
You can also use this feature when the network devices used in H.323-SIP interworking do not include SDP in the INVITEs sent to SIP endpoints. In this case, the Oracle Enterprise Session Border Controller can insert SDP in the outgoing INVITE messages it forwards to the next hop.

This feature works for both INVITEs and ReINVITEs.

This section explains how the SDP insertion feature works for INVITEs and ReINVITEs. The examples used this section are both pure SIP calls. Even when you want to use this feature for IWF calls, though, you configure it for the SIP side.

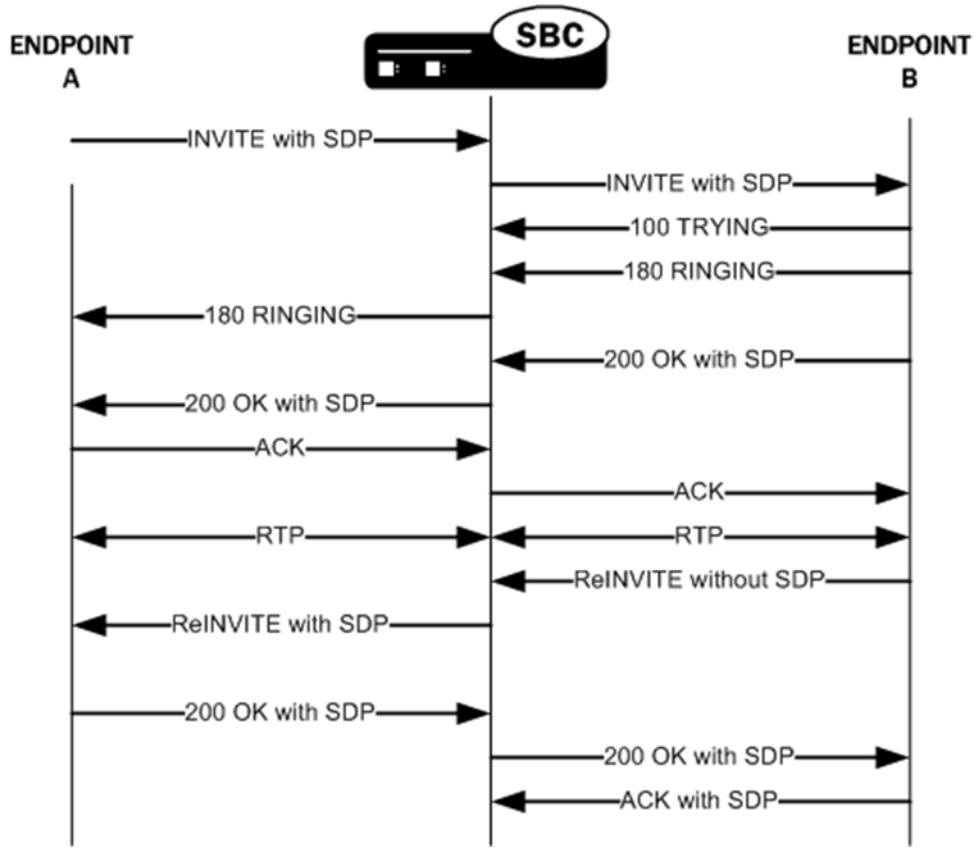
SDP Insertion for SIP INVITES

With the parameters mentioned above appropriately configured, the Oracle Enterprise Session Border Controller inserts SDP into an outgoing INVITE when the corresponding incoming INVITE has none. Because no SDP information is available for the session, the Oracle Enterprise Session Border Controller uses a media profile from a list of them you configure and then apply for SDP insertion.



SDP Insertion for SIP ReINVITEs

The section explains SDP insertion for ReINVITEs, using a case where SIP session has been established with an initial INVITE containing SDP. In the diagram below, you can see the initial INVITE results in a negotiated media stream. But after the media stream is established, Endpoint B sends a ReINVITE without SDP to the Oracle Enterprise Session Border Controller. In this case, the Oracle Enterprise Session Border Controller inserts the negotiated media information from the initial INVITE as the ReINVITE's SDP offer. For subsequent ReINVITEs with no SDP, the Oracle Enterprise Session Border Controller inserts the negotiated media information from the last successful negotiation as the ReINVITE's SDP offer. It then sends this ReINVITE with inserted SDP to the next hop signaling entity.



SDP Insertion Configuration

This section shows you how to configure SDP insertion for the calls cases described above.

Configuring SDP Insertion for SIP INVITEs

To work properly, SDP insertion for SIP invites requires you to set a valid media profile configuration.

To enable SDP insertion for INVITEs:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-config)#
```

4. add-sdp-invite—Change this parameter from disabled (default), and set it to invite.
5. add-sdp-profile—Enter a list of one or more media profile configurations you want to use when the system inserts SDP into incoming INVITEs that have no SDP. The media profile contains media information the Oracle Enterprise Session Border Controller inserts in outgoing INVITE.

   This parameter is empty by default.
6. Save and activate your configuration.

Configuring SDP Insertion for SIP ReINVITEs

In this scenario, the Oracle Enterprise Session Border Controller uses the media information negotiated early in the session to insert after it receives an incoming ReINVITE without SDP. The Oracle Enterprise Session Border Controller then sends the ReINVITE with inserted SDP to the next hop signaling entity. You do not need the media profiles setting for ReINVITEs.

To enable SDP insertion for ReINVITEs:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-config)#
```

4. add-sdp-invite—Change this parameter from disabled (default), and set it to reinvite.
5. Save and activate your configuration.

Restricted Media Latching

The restricted media latching feature lets the Oracle Enterprise Session Border Controller latch only to media from a known source IP address, in order to learn and latch the dynamic UDP port number. The restricting IP address's origin can be either the SDP information or the SIP message's Layer 3 (L3) IP address, depending on the configuration.

About Latching

Latching is when the Oracle Enterprise Session Border Controller listens for the first RTP packet from any source address/port for the destination address/port of the Oracle Enterprise Session Border Controller. The destination address/port is allocated dynamically and sent in the SDP. After it receives a RTP packet for that allocated destination address/port, the Oracle Enterprise Session Border Controller only allows subsequent RTP packets from that same source address/port for that particular Oracle Enterprise Session Border Controller destination address/port. Latching does not imply that the latched source address/port is used for the destination of the reverse direction RTP packet flow (it does not imply the Oracle Enterprise Session Border Controller will perform symmetric RTP).

Restricted Latching

The Oracle Enterprise Session Border Controller restricts latching of RTP/RTCP media for all calls within a realm. It latches to media based on one of the following:

- SDP: the IP address and address range based on the received SDP c= connect address line in the offer and answer.
- Layer 3: the IP address and address range based on the received L3 IP address of the offer or answer. This option is for access registered HNT endpoints. If the L3 IP address is locally known and cached by the Oracle Enterprise Session Border Controller as the public SIP contact address, that information could be used instead of waiting for a response. The Oracle Enterprise Session Border Controller might use the L3 IP address restriction method for all calls regardless of whether the endpoint is behind a NAT or not, for the same realms.

Symmetric Latching

A mode where a device's source address/ports for the RTP/RTCP it sends to the Oracle Enterprise Session Border Controller (E-SBC) that are latched, are then used for the destination of RTP/RTCP sent to the device.

After allocating the media session in SIP, the E-SBC sets the restriction mode and the restriction mask for the calling side as well as for the called side. It sets the source address and address prefix bits in the flow. It also parses and loads the source flow address into the MIBOCO messages. After receiving the calling SDP, the E-SBC sets the source address (address and address prefix) in the appropriate flow (the flow going from calling side to the called side). After receiving the SDP from the called side, the E-SBC sets the source address in the flow going from the called side to the calling side.

The E-SBC uses either the address provided in the SDP or the layer 3 signaling address for latching. You also configure the E-SBC to enable latching so that when it receives the source flow address, it sets the address and prefix in the NAT flow. When the NAT entry is installed, all the values are set correctly. In addition, sipd sends the information for both the incoming and outgoing flows. After receiving SDP from the called side sipd, the E-SBC sends information for both flows to the MBCD so that the correct NAT entries are installed.

Enabling restricted latching may make the E-SBC wait for a SIP/SDP response before latching, if the answerer is in a restricted latching realm. This is necessary because the E-SBC does not usually know what to restrict latching to until the media endpoint is reached. The only exception could be when the endpoint's contact/IP is cached.

Relationship to Symmetric Latching

The current forced HNT symmetric latching feature lets the Oracle Enterprise Session Border Controller assume devices are behind NATs, regardless of their signaled IP/SIP/SDP layer addresses. The Oracle Enterprise Session Border Controller latches on any received RTP destined for the specific IP address/port of the Oracle Enterprise Session Border Controller for the call, and uses the latched source address/port for the reverse flow destination information.

If both restricted latching and symmetric latching are enabled, the Oracle Enterprise Session Border Controller only latches if the source matches the restriction, and the reverse flow will only go to the address/port latched to, and thus the reverse flow will only go to an address of the same restriction.

- Symmetric latching is enabled.

  If symmetric latching is enabled, the Oracle Enterprise Session Border Controller sends the media in the opposite direction to the same IP and port, after it latches to the source address of the media packet.
- Symmetric latching is disabled.

If symmetric latching is disabled, the Oracle Enterprise Session Border Controller only latches the incoming source. The destination of the media in the reverse direction is controlled by the SDP address.

Example 1

A typical example is when the Oracle Enterprise Session Border Controller performs HNT and non-HNT registration access for endpoints. Possibly the SDP might not be correct, specifically if the device is behind a NAT. Therefore the Oracle Enterprise Session Border Controller needs to learn the address for which to restrict the media latching, based on the L3 IP address. If the endpoint is not behind a NAT, then the SDP could be used instead if preferred. However, one can make some assumptions that access-type cases will require registration caching, and the cached fixed contact (the public FW address) could be used instead of waiting for any SDP response.

Example 2

Another example is when a VoIP service is provided using symmetric-latching. A B2BUA/proxy sits between HNT endpoints and the Oracle Enterprise Session Border Controller, and calls do not appear to be behind NATs from the Oracle Enterprise Session Border Controller's perspective. The Oracle Enterprise Session Border Controller's primary role, other than securing softswitches and media gateways, is to provide symmetric latching so that HNT media will work from the endpoints.

To ensure the Oracle Enterprise Session Border Controller's latching mechanism is restricted to the media from the endpoints when the SIP Via and Contact headers are the B2BUA/proxy addresses and not the endpoints', the endpoint's real (public) IP address in the SDP of the offer/answer is used. The B2BUA/proxy corrects the c= line of SDP to that of the endpoints' public FW address.

The Oracle Enterprise Session Border Controller would then restrict the latching to the address in the SDP of the offer from the access realm (for inbound calls) or the SDP answer (for outbound calls).

Restricted Latching Configuration

To configure restricted latching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>           0.0.0.0
2: MGCP_Realm <none>           0.0.0.0
3: H323REALM  <none>           0.0.0.0
selection:1
ACMEPACKET(realm-config)#
```

5. restricted-latching— Enter the restricted latching mode. The default is none. The valid values are:
   - none—No restricted-latching used
   - sdp—Use the address provided in the SDP for latching
   - peer-ip—Use the layer 3 signaling address for latching
6. restriction-mask— Enter the number of address bits you want used for the source latched address. This field will be used only if the restricted-latching is used. The default is 32. When this value is used, the complete IP address is matched for IPv4 addresses. The valid range is:

- Minimum—1
- Maximum—128

Enhanced SIP Port Mapping

This section explains how to configure SIP port mapping feature to support:

- Anonymous requests from endpoints
- Cases where endpoints dynamically change transport protocols between UDP and TCP

Anonymous Requests

If a SIP endpoint sends an INVITE message with a From header that is anonymous, the Oracle Enterprise Session Border Controller can find the registration cache entry by using the Contact and Via headers. In cases such as instant messaging (IM), where there is no Contact header, the Oracle Enterprise Session Border Controller can use the Via header.

The Oracle Enterprise Session Border Controller's checks whether the reg-via-key option is configured for the access-side SIP interface where a REGISTER is received. If the option is enabled, the Oracle Enterprise Session Border Controller makes the via-key by adding the IP address from the Via header to the firewall address (if there is a firewall present between the Oracle Enterprise Session Border Controller and the endpoint).

When an INVITE arrives at a SIP interface where this option is enabled, the Oracle Enterprise Session Border Controller determines whether the From header is anonymous or not. If it is anonymous, then the Oracle Enterprise Session Border Controlleruses the Via-key to find the registration entry.

Anonymous SIP Requests Configuration

To enable support for anonymous SIP requests:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. Type options +reg-via-key and press Enter.

   ```
   ACMEPACKET(sip-interface)# options +reg-via-key
   ```

   If you type options reg-via-key without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your configuration.

SIP Registration Via Proxy

The Oracle Enterprise Session Border Controller supports a number of features that require it to cache registration information for UAs (endpoints) registering and receiving requests through it. For those features to operate correctly, the Oracle Enterprise Session Border Controller must act as the outbound proxy through which these endpoints register.

In order to support deployments where a proxy sits between the Oracle Enterprise Session Border Controller and the endpoints, the Oracle Enterprise Session Border Controller must consider the bottom Via header it receives from

endpoints when constructing and matching cache registration entries. And when you use SIP port mapping, the system must use the bottom Via header as a way to determine the endpoint uniquely so that it can have a unique mapping port when the SIP interface is configured with the reg-via-key=all option.

Using the reg-via-proxy option, you can enable your Oracle Enterprise Session Border Controller to support endpoints that register using an intervening proxy. You can set this option either for a realm or for a SIP interface. If you use it for a SIP interface, add to the SIP interface pointing toward the proxy and endpoints—the access side.

Considerations for Reg-Via-Key and Port Mapping

When you set the reg-via-proxy option, the Oracle Enterprise Session Border Controller includes the bottom Via header from received requests in the registration cache Via Key. The system also uses it for determining whether or not the request matches a registration cache entry. Each unique bottom Via received a unique mapping port when you turn SIP port mapping on and set the SIP interface with the reg-via-key=all option.

Request Routing

So that requests addressed to the corresponding registered contact are routed to the proxy, the Oracle Enterprise Session Border Controller includes the intervening proxy (i.e., the top Via) in the routing information for the registration cache when you set reg-via-proxy. To carry out this routing scheme, the system adds a Path header (if none is present) to the REGISTER. But it removes the Path header prior to sending the REGISTER to the registrar.

Note that when the received REGISTER contains a Path header, the Oracle Enterprise Session Border Controller uses it for routing requests toward the endpoint and includes it in the forwarded REGISTER request—as is the case when you do not enable SIP registration via proxy.

SIP Registration Via Proxy Configuration

To configure SIP registration via proxy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. Type options +reg-via-proxy and press Enter.

   ```
   ACMEPACKET(sip-interface)# options +reg-via-proxy
   ```

   If you type options reg-via-proxy without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your configuration.

Dynamic Transport Protocol Change

The Oracle Enterprise Session Border Controller also uses the IP address and port in the Contact and Via headers. This is useful for cases when endpoints dynamically change transport protocols (TCP/UDP), and the port number used for sending an INVITE might not be the same one used to send a Register message.

If you do not enable this feature, when an endpoint registered with the Oracle Enterprise Session Border Controller used UDP for its transport protocol, a call fails if that endpoint subsequently initiates the call using TCP. The Oracle Enterprise Session Border Controller checks for the Layer 3 IP address and port, and it rejects the call if the port is changed.

With the new option reg-no-port-match added to the SIP interface configuration, the Oracle Enterprise Session Border Controller will not check the Layer 3 port in the INVITE and REGISTER messages.

Dynamic Transport Protocol Change Configuration

To enable dynamic transport protocol change:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Type options +reg-no-port-match and press Enter.

```
ACMEPACKET(sip-interface)# options +reg-no-port-match
```

   If you type options reg-no-port-match without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your configuration.

SIP Privacy Extensions

This section explains how you can configure privacy services to be applied only when the source is trusted and the destination is considered untrusted. (Prior to this release, the Oracle Enterprise Session Border Controller always applied the privacy services, unless the source and the destination were both trusted.)

The Oracle Enterprise Session Border Controller considers all user endpoints and nodes outside the core as untrusted.

The Oracle Enterprise Session Border Controller acts as the boundary device between the trusted platform and the untrusted Internet, to implement privacy requirements. When it receives a message, the Oracle Enterprise Session Border Controller checks whether the source is trusted. It evaluates the level of privacy requested in a Privacy header, if present.

Depending on whether the source is trusted or untrusted, the Oracle Enterprise Session Border Controller can do different things when passing the message to the outgoing side. It also checks whether the destination is trusted.

Privacy Types Supported

The Oracle Enterprise Session Border Controller supports the following Privacy types:

- user: user-level privacy function provided. Any non-essential informational headers are removed, including the Subject, Call-Info, Organization, User-Agent, Reply-To, and In-Reply-To. Possibly the original value of the From header is changed to anonymous.
- header: headers that cannot be set arbitrarily by the user (Contact/Via) are modified. No unnecessary headers that might reveal personal information about the originator of the request are added. (The values modified must be recoverable when further messages in the dialog need to be routed to the originator.)
- id: third-party asserted identity kept private with respect to SIP entities outside the trust domain with which the user authenticated.

The following SIP headers can directly or indirectly reveal identity information about the originator of a message: From, Contact, Reply-To, Via, Call-Info, User-Agent, Organization, Server, Subject, Call-ID, In-Reply-To and Warning.

user

The Oracle Enterprise Session Border Controller supports the Privacy type user. It can remove non-essential information headers that reveal user information by:

- Setting the SIP From header and display information to anonymous
- Removing the Privacy header
- Removing Proxy-Require option tag = privacy (if present)
- Removing the following headers:

  Subject

  Call-Info

  Organization

  User-Agent

  Reply-To

  In-Reply-To

header

The Oracle Enterprise Session Border Controller also supports the Privacy type header. It modifies SIP headers that might reveal the user identity by:

- Stripping the Via header
- Replacing the Contact header
- Stripping Record-Route
- Removing the Privacy header
- Removing Proxy-Require option tag = privacy (if present)

In general, the B2BUA behavior of the Oracle Enterprise Session Border Controller by default provides header privacy for all sessions.

id

The Oracle Enterprise Session Border Controller also supports the Privacy type id. It keeps the Network Asserted Identity private from SIP entities outside the trusted domain by:

- Stripping only P-Asserted-Identity
- Removing the Privacy header and Proxy-Require option-tag = privacy
- Setting the From header to anonymous (for the backward compatibility)

Examples

The following examples show the actions the Oracle Enterprise Session Border Controller performs depending on the source and target of the calls.

Calls from Untrusted Source to Trusted Target

When calls are from an untrusted source to a trusted target and PPI is included in the INVITE sent to IP border elements, the Oracle Enterprise Session Border Controller maps the PPI information to PAI in the outgoing INVITE to the trusted side (even if the Privacy header is set to id or to none). The Privacy and From headers get passed on unchanged.

IP border elements must pass PAI (if received in the ingress INVITE) and the From and Privacy headers to the egress side just as they were received on the ingress side.

The Oracle Enterprise Session Border Controller maps the PPI to PAI by default, if the outgoing side is trusted. To change this behavior, you need to configure the disable-ppi-to-pai option.

Calls from Trusted to Untrusted

When calls are from a trusted source to an untrusted target, and the Privacy header is set to id, the Oracle Enterprise Session Border Controller strips PAI, makes the From header anonymous, and strips the Privacy header.

If the Privacy header is set to none, the Oracle Enterprise Session Border Controller does not change the From header and passes on the Privacy header, if there is one.

Calls from Trusted to Trusted

When calls are going from trusted source to trusted target acting as a peer network border element and PPI is included, the Oracle Enterprise Session Border Controller maps PPI to PAI. The Privacy header remains the same as signaled and the Oracle Enterprise Session Border Controller passes the From header and the PAI without changes.

Configuring SIP Privacy Extensions

Prior to this release the session agent's trust mode provided this functionality. Now you configure SIP interface's trust-mode as none, which means nothing is trusted for this SIP interface.

You also configure the disable-ppi-to-pai parameter disable the changing of the P-Preferred header to the P-Asserted-Identity header, if the outgoing side is trusted.

Trust Mode

To configure the trust mode:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.

4. If configuring an existing interface, enter the select command to select the interface.
5. trust-mode—Select the trust mode for this SIP interface. The default value is all. The valid values are:

   - all—Trust all previous and next hops except untrusted session agents
   - agents-only—Trust only trusted session agents
   - realm-prefix—Trusted only trusted session agents or address matching realm prefix
   - registered—Trust only trusted session agents or registered endpoints
   - none—Trust nothing

6. Save and activate your configuration.

   The following example shows the trust-mode set to none. The remaining SIP interface options are omitted for brevity.

```
sip-interface
        state                           enabled
        realm-id                        access1
        sip-port
                address                         192.168.1.30
                port                            5060
                transport-protocol              UDP
                allow-anonymous                 all
        carriers
        proxy-mode                      Proxy
```

```
        redirect-action
        contact-mode                    maddr
        nat-traversal                   none
        nat-interval                    30
        registration-caching            disabled
        min-reg-expire                  300
        registration-interval           3600
        route-to-registrar              disabled
        teluri-scheme                   disabled
        uri-fqdn-domain
        options
        trust-mode                      none
```

Disabling the PPI to PAI Change

To disable the changing of PPI to PAI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP configuration parameters. To view all sip-config parameters, enter a ? at the system prompt.

4. If configuring an existing SIP configuration, enter the select command to select it.
5. options—Enter disable-ppi-to-pai. If adding to an existing list of options, use a preceding plus (+) sign.

   ```
   options +disable-ppi-to-pai
   ```

6. Save and activate your configuration.

SIP Registration Cache Limiting

Using SIP registration cache limiting for SIP endpoint access deployments, you can restrict the size of the SIP registration cache for the global SIP configuration.

You can implement this feature if you have been seeing issues where, either due to network failure scenarios or incorrect sizing of system capabilities, the Oracle Enterprise Session Border Controller and/or the SIP registrar cannot support the number of registering endpoints. Although the Oracle Enterprise Session Border Controller protects itself and the registrar against SIP REGISTER floods, conditions can still occur where too many legitimate endpoints attempt to register with the registrar via the Oracle Enterprise Session Border Controller.

By enabling SIP registration cache limiting, you restrict the number of legitimate endpoints that can register. The Oracle Enterprise Session Border Controller rejects any endpoints beyond the limit you set. If you do not want to use this feature, simply leave the reg-cache-limit parameter set to its default of 0, meaning there is no limit to the entries in the SIP registration cache.

When you limit the number of registered endpoints allowed in the Oracle Enterprise Session Border Controller's registration cache, the Oracle Enterprise Session Border Controller analyzes each registration before starting to process it. First, the Oracle Enterprise Session Border Controller checks the contact header to determine if it is already in the list of contacts for the user. If it finds the contact in its cache list, the Oracle Enterprise Session Border Controller treats the registration as a refresh; it treats any other headers as new. Note that the Oracle Enterprise Session Border Controller checks the message prior to making any changes to the cache because it must either accept or reject the message as a whole.

The Oracle Enterprise Session Border Controller adds the number of new contacts to the number already present in the cache, and rejects any registration with a contact that would cause it to exceed its limit. Rejection causes the

Oracle Enterprise Session Border Controller to send a response communicating that its registration cache is full. The default response is the 503 Registration DB-Full message, but you can use the SIP response mapping feature to use another message if required.

You can set an option in the global SIP configuration that defines the value in the Retry-After header. The Oracle Enterprise Session Border Controller sends this header as part of its rejection response when the registration cache is full. Another option sets the percentage of the registration cache size which, if exceeded, causes the Oracle Enterprise Session Border Controller to send an alarm.

About Registration Cache Additions Modifications and Removals

When it receives a REGISTER message with new contact information for a user, the Oracle Enterprise Session Border Controller considers it an addition to the cache and augments the number of registration cache entries. Then the Oracle Enterprise Session Border Controller forwards the message to the registrar, and—when and only when the registrar returns both the original and new contacts in the 200 OK—the registration cache count stays the same. However, if the registrar returns only the new contact (making this a case of modification), then the Oracle Enterprise Session Border Controller removes the old contact information and subtracts accordingly from the number of registration cache entries.

Thus the Oracle Enterprise Session Border Controller does not know whether a REGISTER might result in an addition or a modification until it receives a response from the registrar. For this reason, the Oracle Enterprise Session Border Controller first assumes it is to make an addition, and then updates the registration cache and count when it has the necessary information from the registrar.

The registration cache count does not reflect removals during the rejection check because the Oracle Enterprise Session Border Controller ignores registration messages or expires headers with their expires values set to zero when it counts new entries. The fact that removals take place after additions and modifications means that messages which remove one contact while adding another might be rejected. That is, the addition might exceed the registration cache limit before any removal can take place to make room for it.

Registration Cache Alarm Threshold

A percentage of the registration cache limit, the registration cache alarm threshold is a configurable value you can set to trigger an alarm when the registration cache is reaching its limit. When exceeded, this threshold triggers the generation of an alarm and SNMP trap. When registrations fall back beneath the threshold, the Oracle Enterprise Session Border Controller clears the alarm and sends a clear trap.

This alarm is Major in severity, and its text reads as follows:

```
Number of contacts <registration count> has exceeded the registration cache
threshold <threshold %> of <registration cache limit value>.
```

Notes on Surrogate Registration

The Oracle Enterprise Session Border Controller does not, under any circumstances, reject surrogate registrations on the basis of the registration cache limit. However, surrogate registrations generate contacts, and so they do add to the global registration count. In the case where the surrogate registrations add to the registration count to the extent the count exceeds the limit you configure, you will have more registrations in the cache than the configured limit.

Monitoring Information

You can monitor how many entries are in the SIP registration cache using the ACLI show registration command and referring to the Local Contacts statistics.

SIP Registration Cache Limiting Configuration

This section shows you how to configure the registration cache limit, and how to set the options controlling retry times and thresholds for alarm purposes.

To configure SIP registration cache limiting:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

   If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI select command) before making your changes.

4. registration-cache-limit—Set the registration cache limit, or the maximum number of SIP registrations that you want to keep in the registration cache. The minimum and default value for this parameter is 0, and you can set it to a maximum value of 999999999. Leaving this parameter set to 0 means there is no limit on the registration cache (and therefore leaves this feature disabled).

5. options—Set the options parameter by typing options, a Space, the option name reg-cache-lim-retry-after=X (where X is the value added to the Retry-After header) with a plus sign in front of it. This option defaults to 1800, and you can enter values from 0 to 999999999.

   You can configure the alarm threshold option the same way, substituting the option name reg-cache-alarm-thresh=X (where X is the percentage of registration cache limit that triggers an alarm). This option defaults to 95, and you can enter value from 0 to 100.

```
ACMEPACKET(sip-config)# options +reg-cache-lim-retry-after=2500
ACMEPACKET(sip-config)# options +reg-cache-alarm-thresh=90
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

SIP Registration Overload Protection

You can configure your Oracle Enterprise Session Border Controller for SIP Registration overload protection, which augments the Oracle Enterprise Session Border Controller's protection methods. Working with the Oracle Enterprise Session Border Controller's access control and registration caching functions, this new feature guards against benign avalanche restarts. The avalanche is caused by events where many endpoints lose power or connectivity at once, are restored to service, and then flood the Oracle Enterprise Session Border Controller as they attempt to register again.

Normally, the Oracle Enterprise Session Border Controller handles SIP registration by creating a temporary registration cache for the endpoint's address of record (AoR) and forwards the REGISTER request to the registrar. To challenge the endpoint's registration, the registrar sends back either a 401 Unauthorized or 407 Proxy Authorization Required response. When it receives the 401 or 407, the Oracle Enterprise Session Border Controller saves the challenge context in anticipation of receiving a second REGISTER with the endpoint's authentication credentials. The Oracle Enterprise Session Border Controller forwards the second REGISTER (with authentication credentials) to the registrar, and then the registrar confirms registration with a 200 OK. Both REGISTER requests are subject to the system's access control rules, set either for the ingress realm or the ingress session agent. The Oracle Enterprise Session Border Controller also honors the maximum registration sustain rate constraint for session agents; this applies when the incoming REGISTER is from a session agent and the outgoing REGISTER is sent to a session agent.

When you enable SIP Registration overload protection, the Oracle Enterprise Session Border Controller temporarily promotes the endpoint to the trusted level when it receives the 401 or 407 response (to the first REGISTER) from the registrar. This ensures that the second REGISTER (containing authentication credentials) can reach the Oracle Enterprise Session Border Controller. Temporary promotion lasts only for the amount of time remaining before the REGISTER server transaction expires plus the time allotted in the transaction expiration parameter in the SIP

configuration. Before the temporary promotion expires, there is enough time for any necessary retransmissions of the first REGISTER and for the second REGISTER to take place. The following situations might also occur:

- If the Oracle Enterprise Session Border Controller receives a 401 or 407 to the second REGISTER request, it resets its access control level for the endpoint's address to the default level; it then treats additional REGISTER requests from the same context at the default access control level.
- If the Oracle Enterprise Session Border Controller receives a 200 OK response to the REGISTER message, it extends the promotion time to the expiration period for the registration cache.

If the Oracle Enterprise Session Border Controller is able to find the temporary registration cache and the saved challenge context when the second REGISTER arrives, it forwards the REGISTER without checking the maximum registration sustain rate constraint for ingress and egress session agents—thereby ensuring that the REGISTER with authentication credentials is sent to the registrar. So when you use this feature, you should set the maximum registration sustain rate constraint of the session agent (representing the registrar) at half the registrar's maximum registration sustain rate. Additional REGISTER requests with the same challenge context are subject to the maximum registration sustain rate constraint.

SIP Registration Overload Protection Configuration

When you configure this feature, be sure to set the reg-overload-protect option in your global SIP configuration:

To enable SIP Registration overload protection on your Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (reg-overload-protect), and then press Enter.

```
ACMEPACKET(sip-config)# options +reg-overload-protect
```

   If you type either of these options without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

   Note: Note that the sip-config option "cache-challenges" (enabled by default) must not have been disabled for SIP Registration Overload Protection to work properly. If you have disabled cache-challenges, re-evaluate the reason you disabled it. If registration overload protection supersedes your reason for disabling cache-challenges, re-enable the option as shown below.

```
ACMEPACKET(sip-config)# options +cache-challenges=yes
```

   Note that the configuration syntax above is equivalent to the following, which uses the "-" character to remove the option.

```
ACMEPACKET(sip-config)# options -cache-challenges
```

5. Save and activate your configuration.

SIP Request Method Throttling

You can configure throttling mechanisms for SIP INVITEs and REGISTERs using session agent constraints. However, you might want to throttle other types of SIP methods, and for those methods you should use the rate

constraints configuration available both in the session constraints (which you then apply to a SIP interface or a realm) and the session agent configurations.

Oracle recommends you use session agent constraints for session-rate INVITE throttling and registration-rate for REGISTER throttling.

For SIP access deployments, you can configure rate constraints for individual method types along with a set of burst and sustain rates. These constraints can help to avoid overloading the core network. In addition, they restrain the load non-INVITE messages use, thus reserving capacity for INVITE-based sessions and Registrations

When you configure SIP request method throttling, you must exercise care because it is possible to reject in-dialog requests. Therefore, Oracle recommends you do NOT configure constraints—although the configuration allows you to and will not produce error messages or warnings if you set them—for the following SIP method types:

- ACK
- PRACK
- BYE
- INFO
- REFER

However, the Oracle Enterprise Session Border Controller is likely to throttle NOTIFY requests despite their being part of a Subscribe dialog.

Therefore, the methods you will most likely configure for throttling are:

- NOTIFY
- OPTIONS
- MESSAGE
- PUBLISH
- REGISTER

The Oracle Enterprise Session Border Controller counts Re-INVITEs and challenged responses against the throttle limit, but does not check to determine if the constraints have been exceeded for either.

You can configure separate constraints—inbound and outbound values for burst and sustain rates—for each different method type you configure. Although you should use session agent constraints (and not rate constraints) for INVITEs, if you also set up rate constraints for INVITEs, then the smallest configured value takes precedence.

About Counters and Statistics

Each rate constraint you configure for a SIP method tracks its own counters. For example, if you configure a rate constraint for the PUBLISH method, the burst and sustain rates you set for it apply only to the PUBLISH method and not to any other methods for which you might set up rate constraints. You can, however, set the burst rate window in the session constraints configuration that will apply to all methods configured as rate constraints.

The Oracle Enterprise Session Border Controller captures statistics for SIP methods throttled by rate constraints for SIP interfaces and session agents; it does not capture these statistics for the global SIP configuration.

SIP Request Method Throttling Configuration

This section shows you how to set up rate constraints for session constraints (which are then applied to SIP interfaces) and session agents.

To use this feature, you must enable the extra-method-stats parameter in the global SIP configuration.

To set the extra-method-stats parameter in the global SIP configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
If you are adding this feature to an existing configuration, you need to
select the configuration (using the ACLI select command) before making your
changes.
```

4. extra-method-stats—Set this parameter to enabled.
5. Save and activate your configuration.

Rate Constraints for SIP Interfaces

To apply rate constraints to SIP interfaces, you need to configure rate constraints in the session constraints configuration and then apply the session constraints to the SIP interface where you want them used.

Note that you need to set up the parent session-constraint configuration to save any rate constraints you configure.

To configure rate constraints:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-constraints and press Enter.

```
ACMEPACKET(session-router)# session-constraints
ACMEPACKET(session-constraints)#
```

   If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. Type rate-constraints and press Enter.

```
ACMEPACKET(session-constraints)# rate-constraints
ACMEPACKET(rate-constraints)#
```

5. method—Enter the SIP method name for the method you want to throttle. Although the parameter accepts other values, your entries should come only from the from the following list for the feature to function properly:
   - NOTIFY
   - OPTIONS
   - MESSAGE
   - PUBLISH
   - REGISTER
6. max-inbound-burst-rate—For the SIP method you set in the methods parameter, enter the number to restrict the inbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
7. max-outbound-burst-rate—For the SIP method you set in the methods parameter, enter the number to restrict the outbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
8. max-inbound-sustain-rate—For the SIP method you set in the methods parameter, enter the number to restrict the inbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
9. max-outbound-sustain-rate—For the SIP method you set in the methods parameter, enter the number to restrict the outbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.

10. Save your changes and apply this session constraint and its rate constraint(s) to SIP interfaces.

Applying Session and Rate Constraints to a SIP Interface

You need the name of the session constraints configuration to apply the restrictions you set up to a SIP interface.

To apply session and rate constraints to a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

   If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. constraint-name—Enter the name of the session constraint configuration where you have set up rate constraints to apply them to this SIP interface. This parameter has no default, and must be the valid name of a session constraint configuration.
5. Save and activate your configuration.

Configuring Rate Constraints for Session Agents

You can also use this feature for individual SIP session agents.

To configure rate constraints for a SIP session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. Type rate-constraints and press Enter.

```
ACMEPACKET(session-agent)# rate-constraints
ACMEPACKET(rate-constraints)#
```

5. method—Enter the SIP method name for the method you want to throttle. Your entries should come only from the following list:

   - NOTIFY
   - OPTIONS
   - MESSAGE
   - PUBLISH
   - REGISTER

6. max-inbound-burst-rate—For the SIP method you set in the methods parameter, enter the number to restrict the inbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
7. max-outbound-burst-rate—For the SIP method you set in the methods parameter, enter the number to restrict the outbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
8. max-inbound-sustain-rate—For the SIP method you set in the methods parameter, enter the number to restrict the inbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
9. max-outbound-sustain-rate—For the SIP method you set in the methods parameter, enter the number to restrict the outbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
10. Save and activate your configuration.

SIP Delayed Media Update

The Oracle Enterprise Session Border Controller supports SIP delayed media update. When enabled, this feature keeps the Oracle Enterprise Session Border Controller from updating its media flow information for flows established after an offer-answer exchange. The Oracle Enterprise Session Border Controller does not update the flow information until a new offer and answer arrive for a specific set of media flows.

The (subsequent) offer does not have to be for the same session; rather, it can appear as a new SIP INVITE that uses the same SDP.

Delayed Media Update Disabled

When this feature is disabled (which is the default behavior), the Oracle Enterprise Session Border Controller updates media flow entries in its CAM based on signaled SDP when it processes the SDP. If it processes an SDP offer, Oracle Enterprise Session Border Controller allocates steering port resources; the Oracle Enterprise Session Border Controller updates any missing elements for the flow when the answer is returned.

In cases when a secondary offer arrives (either a reINVITE, an UPDATE, or the original INVITE is hairpinned back through the Oracle Enterprise Session Border Controller), the Oracle Enterprise Session Border Controller updates the following media flow information at the time of the offer

- Destination IP address
- Destination port
- Realm for the media flows
- Media release settings

This behavior affects specific applications that are better served by the Oracle Enterprise Session Border Controller waiting to update media flow information until it receives the answer to the second offer.

Delayed Media Update Enabled

When you enable the SIP delayed media update feature, the Oracle Enterprise Session Border Controller:

- Delays changing the active media flow CAM entry for a new offer if a previous offer and answer have been received for the same media flows; it encodes new SDP information in an outgoing offer, but does not change the CAM entry until the answer is received
- Delays changing the active media flow CAM entry even when the new offer is for a new session
- Supports media release when performing delayed media update changes
- Offers per-realm configuration

This section describes how the delayed media update feature works for hairpinned call flows and for an SDP offer arriving for installed flows.

- Hairpinned call flows—In this type of call flow, the application server (AS) sends an INVITE back to the Oracle Enterprise Session Border Controller and that INVITE needs to be forwarded to another user (user B). When it receives the offer in this INVITE and delayed media update is disabled, the Oracle Enterprise Session Border Controller determines that the call is hairpinned and deletes the CAM entry for the flow for user A, who has sent the initial INVITE. The Oracle Enterprise Session Border Controller deletes the CAM entry for the flow from the AS to user A.

  With delayed media update enabled, the CAM entry for the flow from the AS to user A is not deleted. Instead, the Oracle Enterprise Session Border Controller waits until it has an answer from user B, and then performs the necessary updates and deletions.
- SDP offer for installed media flows—With delayed media update enabled, if it has received an offer and answer and a new offer arrives for the same flow, the Oracle Enterprise Session Border Controller delays updating the CAM entries until an answer is received for the new offer.

SIP Delayed Media Update Configuration

You enable this feature on a per-realm basis by setting one parameter.

To enable SIP delayed media update:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type media-manager and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```
3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.
4. delay-media-update—Enable keeping the Oracle Enterprise Session Border Controller from updating its media flow information for flows established after an offer/answer exchange. The default is disabled. The valid values are:

   - enabled | disabled
5. Save and activate your configuration.

Expedited Call Leg Release for Preempted Hairpin Calls

When hairpinned calls are ended because of signaling failures (such as a SIP mid-dialog signaling timeout, or an H.323 TCP keepalive failure) on one call leg, the Oracle Enterprise Session Border Controller deletes both legs' media flows simultaneously by default. In addition, when the first hairpinned call leg is torn down, the second call leg is gracefully released immediately by the Oracle Enterprise Session Border Controller creating and sending an appropriate signaling message (e.g., BYE for a SIP call or ReleaseComplete for an H.323 call) to the endpoint.

You can override this behavior by configuring the dont-terminate-assoc-legs option in the media manager. When configured, the orphaned call leg in the hairpin scenario will be torn down after the initial guard timer expires. The disconnect times of the two call legs, as recorded by the accounting appliation, will be significantly different, due to the initial guard time for the second call leg.

Accounting Considerations

To indicate cases like this, where the second leg of the hairpinned call was preempted, Oracle Enterprise Session Border Controller includes the following combination of release/termination causes in the CDR:

```
VSA 49: Acct-Terminate-Cause = NAS_REQUEST
VSA 62: Acme-Disconnect-Cause = 8
```

SIPconnect

The Oracle Enterprise Session Border Controller supports the SIPconnect model, wherein PBXs register themselves so that service providers do not need to know IP addresses or locations in advance for static configurations. This is particularly helpful when the PBX is behind a NAT.

In the PBX registration process, the PBX creates a binding between one of its phone numbers as the address of record (AoR) and Contact-URI in the REGISTER message. The registrar knows that the single AoR actually represents many addresses, and so it registers them implicitly. However, the registrar does not return the implicit AoR number in P-Associated-URIs.

The SIPconnect feature resolves the following issues that arise from using this model:

- SIP INVITEs sent to the PBX from the Registrar through the Oracle Enterprise Session Border Controller have the Request-URI of registered contact. Because it typically ignores the To-URI, the PBX needs the Request-URI username portion to be the specific extension number being called.

  With the SIP connect feature enabled, the Oracle Enterprise Session Border Controller overwrites the Request-URI username with the To-URI username.
- SIP INVITEs from the PBX have the From AoR and Contact-URI usernames of specific phones rather than of the registered AoR and Contact-URI. For the Oracle Enterprise Session Border Controller, this means that it cannot use the allow-anonymous parameter value of register; there would be no registered user matches, and the Oracle Enterprise Session Border Controller would reject them (with a 403 Forbidden).

  With the SIP connect feature enabled, the Oracle Enterprise Session Border Controller performs allow-anonymous checking based on the registered Via address, which is the same for all requests for the same PBX.

Modifications to Registration Caching Behavior

With the SIP connect feature enabled, Oracle Enterprise Session Border Controller registration caching works the same way that it does with the feature disabled, with the following exceptions:

The Oracle Enterprise Session Border Controller determines whether the destination realm has the sip-connect-pbx-reg option configured, and then:

- If it is configured, the Oracle Enterprise Session Border Controller replaces the user part of the Request-URI with the user part of the To header. When the INVITE contains a P-Called-Party-ID header, the Oracle Enterprise Session Border Controlleruses the user part of the P-Called-Party-ID header (instead of the To header).
- If it is not configured, the Oracle Enterprise Session Border Controller determines if the destination address is for a session agent and whether that session agent has sip-connect-pbx-reg option configured. When it is configured, the Oracle Enterprise Session Border Controller performs the same replacements described in the bullet directly above. When it is not configured, the Oracle Enterprise Session Border Controller does not make any replacements.

When it receives an INVITE request, the Oracle Enterprise Session Border Controller checks the incoming realm for the sip-connect-pbx-reg option.

- If it is configured, the Oracle Enterprise Session Border ControllerC uses the INVITE's source address (instead of the AoR and Contact-URI) to search the registration cache for a matched registration entry.
- If it is not configured, the Oracle Enterprise Session Border Controller determines if the INVITE's source address is for a session agent and whether that session agent has sip-connect-pbx-reg option configured.

  When it is configured, the Oracle Enterprise Session Border Controller replaces the user part of the Request-URI with the user part of the To header. When the INVITE contains a P-Called-Party-ID header, the Oracle Enterprise Session Border Controlleruses the user part of the P-Called-Party-ID header (instead of the To header).

  When it is not configured, the Oracle Enterprise Session Border Controller does not make any replacements.

Configuring SIP Connect Support

You configure this feature by adding the sip-connect-pbx-reg option to the realm configuration. In addition, though this feature requires that your configuration also be set up as outlined in this section. The first two items are required, and Acme Packet recommends that you also implement the suggested additional configuration.

Required Configuration

- Registration caching is enabled.
- For the realm from which registrations come, the options list must include sip-connect-pbx-reg; this is new configuration introduced to support this feature. The presence of this option instructs the Oracle Enterprise Session Border Controller to skip matching the Contact header in the INVITE request with the registered Contact of the registration entry. The Oracle Enterprise Session Border Controller finds a registration using only the INVITE's source address.

  Alternatively, you can configure the sip-connect-pbx-reg option in the options list for a session agent. When the realm where an INVITE comes from does not have this option set, the Oracle Enterprise Session Border Controller determines whether or not the INVITE came from a session agent. You might choose to configure session agents with this option if you do not want it applied to an entire realm. If the PBX is behind a NAT device, the session agent's IP address for the PBX (if statically configured) must be the IP address of the NAT device. And if DNS is use, the session agent's hostname must resolve to the NAT device's IP address.

Suggested Additional Configuration

- In the SIP ports configuration (accessed through the SIP interface configuration), the allow-anonymous parameter must be set to registered. This setting allows the Oracle Enterprise Session Border Controller to accept SIP requests from session agents and registered endpoints only, but to accept REGISTER requests from any endpoint.
- For the SIP interface that accepts registrations, the options parameter must be set to reg-via-key. This setting allows the Oracle Enterprise Session Border Controller to use the source address of an INVITE as the key to find a registration entry in the registration cache. When the INVITE's Contact header matches the registered Contact in the registration entry, the Oracle Enterprise Session Border Controller accepts the INVITE request.

SIP Connect Configuration

To set the SIP connect option for a realm configuration:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. options—Set the options parameter by typing options, a Space, the option name sip-connect-pbx-reg with a plus sign in front of it, and then press Enter.

   ```
   ACMEPACKET(realm-config)# options +sip-connect-pbx-reg
   ```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

   To set the SIP connect option for a SIP session agent configuration:

6. In Superuser mode, type configure terminal and press Enter.

   ```
   Oracle Enterprise Session Border Controller# configure terminal
   ```

7. Type session-router and press Enter to access the signaling-related configurations.

```
Oracle Enterprise Session Border Controller(configure)# session-router
```

8. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the session agent that you want to edit.

9. options—Set the options parameter by typing options, a Space, the option name sip-connect-pbx-reg with a plus sign in front of it, and then press Enter.

```
Oracle Enterprise Session Border Controller(session-agent)# options +sip-
connect-pbx-reg
```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the session agent's configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

10. Save and activate your configuration.

SIP Registration Event Package Support

Certain endpoints subscribe to the Registration Event Package, RFC 3680, which defines how SIP user agents can request and obtain notifications about registration events. Previously, the Oracle Enterprise Session Border Controller passed the Subscribe and Notify messages of this package transparently, without modifying the XML bodies of either. However, in many cases the XML body can contain IP addresses, contact URIs, and expires times that the Oracle Enterprise Session Border Controller needs to modify for proper operation. This new feature enables the Oracle Enterprise Session Border Controller to modify correctly the XML body for the Registration Event Package.

In addition to resolving this type of issue, enabling registration event package support on your system provides the functions described below:

- The Oracle Enterprise Session Border Controller performs NAT on all contacts in the reginfo, regardless of their state.
- The Oracle Enterprise Session Border Controller performs NAT on the address of record (AoR) attribute of the Registration element when it matches an existing cache entry. When either the Contact-URI or the AoR does not match a cache entry and the host part of the URI is an IP address, the Oracle Enterprise Session Border Controller will NAT the host part using the applicable SIP NAT configuration
- Contacts are found in the XML URI element for the contact. But if there is no URI element, then the Oracle Enterprise Session Border Controller uses the Contact element information for the contact.
- If the expires attribute in the Contact element is a value other than zero, the Oracle Enterprise Session Border Controller uses (inserts) the expires values from the registration cache.
- This feature also introduces delayed deletion from the registry cache. When a 200 OK comes back in response to a REGISTER message and the 200 OK does not include all previously registered contacts, the missing contacts are deleted. If the global SIP configuration option contact_cache_linger=XX (where XX is the number of seconds to wait before deleting), then the contacts to be deleted remain for the specified number of seconds before they in fact are deleted.

Updating Expiration Values

This feature also supports updating the expiration values for the registration cache when a Contact element has the expires attribute. For this support, the following apply:

- If the value of the expires attribute is greater than the expiration value for the access-side registration cache entry, the Oracle Enterprise Session Border Controller replaces the XML expires attribute value with the cached one from the access side.
- If the value of the XML expires attribute is less than the core-side expiration value for the core-side registration cache entry, the Oracle Enterprise Session Border Controller updates the core-side expiration value with the value

from the expires attribute. Further, the Oracle Enterprise Session Border Controller adjusts the access-side expiration value of the registration cache in these ways:

  - If the value of the XML expires attribute is less than the current access-side expiration value for the registration cache entry, the Oracle Enterprise Session Border Controller sets the access-side expiration value to be equal to the value in the expires attribute.
  - Otherwise, the Oracle Enterprise Session Border Controller leaves the expires value for the access-side expiration value for the registration cache entry unchanged. If this happens, the Oracle Enterprise Session Border Controller replaces the value of the XML expires attribute with the adjusted access-side expiration value.

- If the expires attribute from a Contact element is 0 (meaning that the core is removing the registration), the Oracle Enterprise Session Border Controller removes that Contact-URI from its registration cache. And if the registration cache entry has no remaining Contact-URIs, the Oracle Enterprise Session Border Controller deletes the registration cache entry altogether.

Contact Cache Linger Configuration

You enable this feature as part of the global SIP configuration, using that configuration's options parameter. You can optionally configure the number of seconds you want to keep a contact in the registration cache before it is deleted. This is the option:

- contact-cache-linger=XX—Number of seconds to wait before a contact is deleted from the cache (where XX is the number of seconds)

To enable SIP Registration overload protection on your Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (contact-cache-linger=XX) where XX is the number of seconds to keep a contact in the cache before deleting it.

```
ACMEPACKET(sip-config)# options +contact-cache-linger=5
```

   If you type either of these options without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

5. Save and activate your configuration.

SIP Event Package for Registrations

This feature enables the Oracle Enterprise Session Border Controller, acting as a Proxy Call Session Control Function (P-CSCF) to initiate subscription to the SIP Event Package for Registrations. Support for the SIP Event Package for Registrations requires no special license.

Applicable Standards

RFC 3265, *Session Initiation Protocol (SIP)-Specific Event Notification*, outlines a framework within which a SIP node, designated as the subscriber, can request automatic, asynchronous notification of certain events from a remote peer, designated as the notifier. RFC 3265 also defines an Event Package as a set of state information to be reported by a notifier to a subscriber, and mandates that such Event Packages be described in a specific RFC explicitly

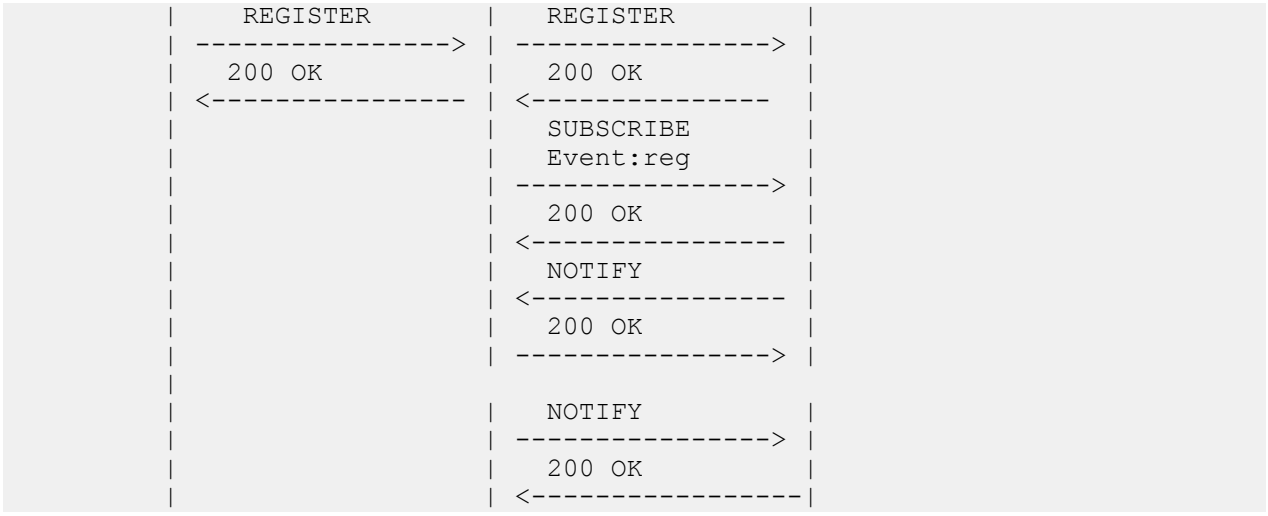
identifying the state information to be reported by the notifier and defining required syntax and semantics to support the subscription/notification exchange.

RFC 3680, *A Session Initiation Protocol (SIP) Event Package for Registrations*, fulfills this requirement by defining a method that enables SIP user agents to request a defined set of state information from a SIP Registrar.

Section 5.2.3 of the 3GPP (Third Generation Partnership Project) 24.229, *IP Multimedia Call Control Protocol Based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)*; Stage 3, mandates that a P-CSCF subscribe to the SIP Event Package for Registrations as defined in RFC 3680.

Call Flow

An example call flow between a SIP endpoint, the Oracle Enterprise Session Border Controller, acting as a P-CSCF, and a SIP Registrar (S-CSCF) illustrates the Subscription/Notification process.



The first two messages (the REGISTER request and the 200 REGISTER response) accomplish the successful registration of the SIP endpoint.

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnaaff
From: sip:joe@example.com;tag=99a8s
To: sip:joe@example.com
Call-ID: 88askjda9@pc34.example.com
CSeq: 9976 REGISTER
Contact: sip:joe@pc34.example.com
```

Immediately after processing the initial 200 OK from the SIP Registrar, the Oracle Enterprise Session Border Controller sends a SUBSCRIBE Request to the S-CSCF.

```
SUBSCRIBE sip:joe@example.com SIP/2.0
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7
From: sip:sd.example.com;tag=123aa9
To: sip:joe@example.com
Call-ID: 9987@app.example.com
CSeq: 9887 SUBSCRIBE
Contact: sip:joe@pc34.example.com
P-Asserted-Identity: <sip:sd@example.com>
Event: reg
Max-Forwards: 70
Accept: application/reginfo+xml
```

The Request URI and To header contain the address-of-record (sip:joe@example.com) of the subscription subject. This value was previously contained in the From and To headers of the of the original REGISTER request. These fields are always identical in REGISTER requests, except in the case of third-party registration.

The From and P-Asserted-Identity headers contain the identity of the subscription requester.

The Contact header contains an IP address or FQDN at which the subscription subject can be reached. This field duplicates the value of the Contact header in the original REGISTER request. Multiple contacts can be registered for a single address-of-record.

The Event header contains the required value, reg, which identifies the requested Event Package subscription. reg specifies the SIP Event Package for Registrations.

The Accept header contains the required, default value, application/reginfo+xml, indicating syntactical support for Registration notifications and attached XML notification bodies.

Assuming the S-CSCF accepts the subscription, it responds with a 200 SUBSCRIBE response.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7
;received=192.0.2.1
From: sip:sd.example.com;tag=123aa9
To: sip:joe@example.com;tag=xyzygg
Call-ID: 9987@app.example.com
CSeq: 9987 SUBSCRIBE
Contact: sip: joe@pc34.example.com
Expires: 3600
```

The From header contains the identity of the subscription requester.

The To header contains the address-of-record of the subscription subject.

The Contact header contains an IP address or FQDN at which the subscription subject can be reached.

The Expires header contains the subscription duration in seconds. Upon receipt of a 2xx response to the SUBSCRIBE request, the Oracle Enterprise Session Border Controller stores the information for the established dialog and the expiration time. If continued subscription is required, the Oracle Enterprise Session Border Controller automatically refreshes the subscription to the SIP Event Package for Registrations, either 600 seconds before the expiration time if the initial subscription was for greater than 1200 seconds, or when half of the time has expired if the initial subscription was for 1200 seconds or less.

Following the 200 SUBSCRIBE response, the S-CSCF generates an initial notification, with Event Package state information contained in an XML body.

```
NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaij
From: sip:app@example.com;tag=xyzygg
To: sip:sd.example.com;tag=123aa9
Call-ID: 9987@app.example.com
CSeq: 1289 NOTIFY
Contact: sip:server19.example.com
Event: reg
Max-Forwards: 70
Content-Type: application/reginfo+xml
Content-Length: ...

<?xml version="1.0"?>
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo" version="1"
state="partial">
    <registration aor="sip:joe@example.com" id="a7" state="active">
        <contact id="76" state="init" event="registered"
        duration-registered="0">
            <uri>sip:joe@pc34.example.com</uri>
        </contact>
    </registration>
</reginfo>
```

Notification Bodies

Registration state changes are reported in XML attachments to NOTIFICATIONS generated by the S-CSCF. As shown above, the XML consists of one or more registration elements that report the state of a specific address-of-record. Attributes supported by the registration element are as follows:

aor contains the address-of-record

id identifies this specific registration

state reports the Registration state — init, active, or terminated

init — address-of-record not yet cached

active — address-of-record maintained in current cache

terminated — address-of-record removed from cache, not currently valid

registration elements, in turn, contain one or more child contact elements. Attributes supported by the contact element are as follows>

id identifies this specific contact

state reports the Contact state — active or terminated

event reports the event that generated the last state change — registered, created, refreshed, shortened, expired, deactivated, probation, unregistered, or rejected

duration-registered reports the length (in seconds) of the current registration

contact elements, contain a single child uri element that identifies the contact address of FQDN.

SIP Event Package for Registrations Configuration

Subscription to the SIP Event Package for Registrations is enabled at the SIP interface level.

1. Use the following command sequence to move to sip-interface Configuration Mode.

   ACMEPACKET# configure terminal

   ACMEPACKET(configure)# session-router

   ACMEPACKET(session-router)# sip-interface

   ACMEPACKET(sip-interface)#
2. Use the subscribe-reg-event parameter to subscribe to the SIP Event Package for Registrations.

   By default, subscription is disabled.

   ACMEPACKET(sip-interface)# subscribe-reg-event enabled

   ACMEPACKET(sip-interface)#
3. Use done, exit, and verify-config to complete enabling the Event Package subscription.

SIP Transport Selection

With this feature enabled, when the Oracle Enterprise Session Border Controller forwards a message larger than the value specified in the maximum UDP length parameter, it attempts to open on outgoing TCP connection to do so. This connection might fail for a number of reasons; for example, an endpoint might not support UDP, or it might be behind a firewall. The UDP fallback option addresses this condition. If it is configured in SIP interfaces associated with an outgoing message and a TCP session cannot be established, the Oracle Enterprise Session Border Controller falls back to UDP and transmits the message. When the option is not present, the Oracle Enterprise Session Border Controller's default behavior is to return the SIP status message 513 Message too Large.

SIP Transport Selection Configuration

You enable this feature per SIP interface by setting options that control the maximum UDP length and allow UDP fallback:

- max-udp-length=X (where X is the maximum length)—Sets the largest UDP packers that the Oracle Enterprise Session Border Controller will pass. Packets exceeding this length trigger the establishment of an outgoing TCP session to deliver the packet; this margin is defined in RFC 3261. The system default for the maximum UDP packet length is 1500.

  You can set the global SIP configuration's max-udp-length=X option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.
- udp-fallback—When a request needs to be sent out on the SIP interface for which you have configured this option, the Oracle Enterprise Session Border Controller first tries to send it over TCP. If the SIP endpoint does not support TCP, however, then the Oracle Enterprise Session Border Controller falls back to UDP and tries the request again.

  To enable SIP Transport Selection:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type session-router and press Enter to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```
3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ```
4. options—Set the options parameter by typing options, a Space, the option name max-udp-length=X (where X is the maximum UDP length you want to set), and then press Enter.

   ```
   ACMEPACKET(sip-interface)# options +max-udp-length=900
   ```

   If you type options max-udp-length=X, you will overwrite any previously configured options. In order to append the new option to the sip-interface's options list, you must prepend the new option with a plus sign as shown in the previous example.
5. options—Set the options parameter by typing options, a Space, the option name udp-fallback, and then press Enter.

   ```
   ACMEPACKET(sip-interface)# options +udp-fallback
   ```

   If you type options udp-fallback, you will overwrite any previously configured options. In order to append the new option to the sip-interface's options list, you must prepend the new option with a plus sign as shown in the previous example.
6. Save and activate your configuration.

SIP Method-Transaction Statistic Enhancements

In prior releases, the Oracle Enterprise Session Border Controller tracks SIP session agents, SIP interfaces and SIP realms on a global level. Only counters that are related to session rates and constraints are displayed.

You can now enable your Oracle Enterprise Session Border Controller to track transaction messages for specific SIP session agents, SIP realms, and SIP interfaces.

The following SIP methods are tracked for Recent, Total, and Period Max values:

- INVITE | ACK | BYE | REGISTER | CANCEL | PRACK | OPTIONS | INFO | SUBSCRIBE | NOTIFY | REFER | UPDATE | MESSAGE | PUBLISH | other (unknown)

With this new tracking enhancement, the show sipd command has been updated with a new method argument which allows you to query statistics for a particular method for a given SIP agent, SIP interface, or SIP realm.

SIP Method Tracking Enhancements Configuration

To enable or disable the expanded SIP Method statistics tracking:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
```

4. extra-method-stats—Enable this parameter if you want to use the expanded SIP Method tracking feature. The default is disabled. The valid values are:

   - enabled | disabled

5. Save and activate your configuration.

SIP TCP Connection Reuse

You can configure your Oracle Enterprise Session Border Controller to reuse TCP connections created by SIP peering devices for outgoing SIP in-dialog and out-of-dialog request transactions.

The SIP draft draft-ietf-sip-connect-reuse-07.txt describes a way for SIP UAs to reuse connections created by a remote endpoint for outgoing requests for TLS. The Oracle Enterprise Session Border Controller does not support the model connection reuse signalled by a parameter; rather, it is provisioned on a per-session-agent basis.

You enable SIP TCP connection reuse on a per-session-agent basis. The Oracle Enterprise Session Border Controller checks incoming TCP connection request to determine if they are from session agent that has this feature turned on. When it is, the Oracle Enterprise Session Border Controller adds the connection's source address to its list of alias connections. This is a list of connections that the Oracle Enterprise Session Border Controller can use for outgoing requests rather than creating its own connection (as it does when this feature is not enabled). So if a preferred connection fails, the Oracle Enterprise Session Border Controller can refer to this list and use the alias connection.

The presence of an alias parameter in the Via header is just one mechanism that will call the Oracle Enterprise Session Border Controller to use the inbound TCP/TLS connection for outbound requests. The Oracle Enterprise Session Border Controller will automatically add an alias for the inbound connections in the following circumstances:

- The other end of the connection is behind a NAT. When the Oracle Enterprise Session Border Controller sees that the Via sent-by does not match the source address of the connection, it will automatically reuse the connection to deliver requests to the UA.
- The Contact address of a REGISTER request received on a TCP connection matches the source address and port. This is because the contact adress is the ephemeral port the UA used to form the connection to the Oracle Enterprise Session Border Controller and, therefore, will not be listening on that port for inbound connections.
- The presence of reuse-connections in the options field of the sip-interface will cause the Oracle Enterprise Session Border Controller to reuse all inbound TCP connections for sending rquests to the connected UA.

SIP TCP Connection Reuse Configuration

This section describes how to enable SIP TCP connection reuse for a session agent. Currently there are two options for the new reuse-connections parameter: none (which turns the feature off) and tcp (which enables the feature for TCP connections). You also set the re-connection interval.

To enable SIP TCP connection reuse for a session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the session agent that you want to edit.

4. reuse-connections—Enable or disable SIP TCP connection reuse. The default is none. This value disables the feature. The valid values are:
   - tcp | none
5. tcp-reconn-interval—Enter the amount of time in seconds before retrying a TCP connection. The default for this parameter is 0. The valid range is:
   - Minimum—0, 2
   - Maximum—300
6. Save and activate your configuration.

SIP TCP Keepalive

The Oracle Enterprise Session Border Controller supports a special TCP keepalive mechanism for SIP. By enabling this feature either for a session agent or for a SIP interface, you allow the Oracle Enterprise Session Border Controller to use standard keepalive probes to determine whether or not connectivity with a remote peer has been lost.

This feature adds to the Oracle Enterprise Session Border Controller's pre-existing TCP keepalive functionality that you can enable in the network parameters configuration. Using existing functionality, you can customize keepalive timing by:

- Specifying the number of unacknowledged packets the Oracle Enterprise Session Border Controller sends to the remote peer before it terminates the TCP connection.
- Specifying the number of seconds of idle time before TCP keepalive messages are sent to the remote peer.

You can now set three modes for TCP keepalive for session agents and SIP interfaces:

- none—(Default) Keepalives are not enabled for use with the session agent/SIP interface; when you select this setting for a session agent, it will use the setting for this feature from the SIP interface.
- enabled—Keepalives are enabled for the session agent/SIP interface.
- disabled—Keepalives are disabled for the session agent/SIP interface.

Note that the setting for this feature for a session agent takes precedence over that for a SIP interface. In addition, the session agent offers you a way to set the re-connection interval.

SIP TCP Keepalive Configuration for Session Agents

To enable SIP TCP keepalive for session agents:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the session agent that you want to edit.

4. tcp-keepalive—Enable or disable standard keepalive probes to determine whether or not connectivity with a remote peer is lost. The default value is none. The valid values are:

   - none | enabled | disabled

   ```
   ACMEPACKET(session-agent)# tcp-keepalive enabled
   ```

5. Save and activate your configuration.

SIP TCP Keepalive Configuration for SIP Interfaces

To enable SIP TCP keepalive for SIP interfaces:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter to access the signaling-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-interface and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI select command) the SIP interface that you want to edit.

4. tcp-keepalive—Enable or disable SIP TCP keepalive. The default value is none. The valid values are:

   - none | enabled | disabled

   ```
   ACMEPACKET(session-agent)# tcp-keepalive enabled
   ```

5. Save and activate your configuration.

SIP Enforcement Profile and Allowed Methods

For this feature, you use a configuration called an enforcement profile that allows you to configure sets of SIP methods that you want applied to: the global SIP configuration, a SIP interface, a realm, or a SIP session agent. The enforcement profile is a named list of allowed methods that you configure and then reference from the configuration where you want those methods applied.

SIP Enforcement Profile Configuration

To use the enforcement profile, you need configure it with a name and the list of SIP methods you want to designate as allowed. Then you need to configure the global SIP configuration, a SIP interface, a realm, or SIP session agent to use the set.

Setting Up and Enforcement Profile

To set up an enforcement profile:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type enforcement-profile and press Enter.

```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```

4. name—Enter the name for the enforcement profile. This parameter has no default, but you must note it so that you can apply this set of allowed SIP headers in: the global SIP configuration, a SIP interface, a realm, or SIP session agent.

```
ACMEPACKET(enforcement-profile)# name EnfProfile1
```

5. allowed-methods—Enter a list of SIP methods that you want to allow for this set. The default value is none. Valid values are:

   - INVITE | REGISTER | PRACK | OPTIONS | INFO | SUBSCRIBE | NOTIFY | REFER | UPDATE | MESSAGE | PUBLISH

     To enter multiple methods for the list, type the parameter name followed by a space, then the names of all methods you want to include each separated by a only a comma and in capital letters.

     ```
     ACMEPACKET(enforcement-profile)# allowed-methods INVITE,REGISTER,PRACK
     ```

6. Save and activate your configuration.

Applying an Enforcement Profile

You can apply an enforcement profile to: the global SIP configuration, a SIP interface, a realm, or SIP session agent. This section shows you how to do all four. Remember that if you are adding this functionality to a pre-existing configuration, you need to select the configuration you want to edit.

To apply an enforcement profile to the global SIP configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. enforcement-profile—Enter the name of the enforcement profile you want to apply to the global SIP configuration.
5. Save and activate your configuration.

To apply an enforcement profile to a SIP interface:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

9. enforcement-profile—Enter the name of the enforcement profile you want to apply to this SIP interface.
10. Save and activate your configuration.

To apply an enforcement profile to a SIP session agent:

11. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

12. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

13. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

14. enforcement-profile—Enter the name of the enforcement profile you want to apply to this session agent.
15. Save and activate your configuration.

To apply an enforcement profile to a realm:

16. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

17. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

18. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

19. enforcement-profile—Enter the name of the enforcement profile you want to apply to this realm.
20. Save and activate your configuration.

Enforcement Profile Configuration with subscribe-event

This section shows you how to configure an enforcement profile with a subscribe-event configuration. Remember that you can set up multiple subscribe-event configurations to correspond with the event types you want to control. It also shows you how to apply these limitations to a realm.

Setting Up Subscribe Dialog Limits

Setting up subscribe dialog limits means setting up an enforcement profile. For the sole purpose of setting up the subscription event limits, you only need to configure the name parameters and then as many subscribe-event configurations as you require. The enforcement profile has other uses, such as SIP SDP address correlation, so only configure the parameters you need.

To configure subscribe dialog limits:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type enforcement-profile and press Enter.

```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement profile)#
```

4. name—Enter a name for this enforcement profile. You will use this name later when you apply the enforcement profile to a realm; it is the value you enter into the enforcement-profile parameter in the realm configuration.
5. Still in the enforcement profile configuration, type subscribe-event and press Enter.

```
ACMEPACKET(enforcement profile)# subscribe-event
ACMEPACKET(subscribe-event)#
```

6. event-type—Enter the SIP subscription event type for which you want to set up limits. You can also wildcard this value (meaning that this limit is applied to all event types except the others specifically configured in this enforcement profile). To use the wildcard, enter an asterisk (*) for the parameter value.

   By default, this parameter is blank.

   Note: The value you enter must be configured as an exact match of the event type expected in the SIP messages (except for the wildcard). Further, the value conforms to the event type BNF specified in RFC 3265.

7. max-subscriptions—Enter the maximum number of subscriptions allowed to a user for the SIP subscription event type you entered in the event-type parameter. Leaving this parameter set to 0 (default) means that there is no limit. You can set this parameter to a maximum value of 65535.
8. If you are entering multiple subscribe-event configurations, then you save them each by using the ACLI done command and then repeat Steps 6 and 7 to configure a new one. If you do not save each, then you will simply overwrite the first configuration repeatedly.

   ```
   ACMEPACKET(subscribe-event)# done
   ```

9. When you finish setting up subscribe-event configurations and have saved them, exit to return to the enforcement profile configuration.

   ```
   ACMEPACKET(subscribe-event)# exit
   ```

10. You also need to save the enforcement profile configuration.

    ```
    ACMEPACKET(enforcement profile)# done
    ```

Applying an Enforcement Profile to a Realm

For the Oracle Enterprise Session Border Controller to use the limits you have set up, you need to apply them to a realm.

To apply an enforcement profile to a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type realm-config and press Enter. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. enforcement-profile—Enter the name of the enforcement profile you want to apply to this realm. This value corresponds to the name parameter in the enforcement profile configuration. This parameter has no default value.
5. Save and activate your configuration.

P-Certificate-Subject-Common-Name to REGISTER Messages

Most Enterprises use revocation servers to authenticate certificates when user equipment registers with the Oracle Enterprise Session Border Controller. For high security enterprises, such as government organizations, user equipment, such as a cell phone, may have a certificate installed. If the user equipment is stolen, for example, the thief could use the equipment to register with theOracle Enterprise Session Border Controller and logon to the system before the certificate is revoked from the server.

The Oracle Enterprise Session Border Controller allows you to enable or disable the addition of a User certificate in the incoming REGISTER message header. This provides an additional layer of security when the user equipment

registers with the Oracle Enterprise Session Border Controller. When the feature is enabled, the individual user certificate must match the user's identity during Registration.

You can enable or disable this feature using the "verify-certificate-info-register" parameter under the existing enforcement-profile object in session-router. in the ACLI. When enabled, and a REGISTER message is encountered, the Oracle Enterprise Session Border Controller adds the User certificate information to the message header. The header is then used in validating the Request-URI Based on certificate information.

Configure the P-Certificate-Subject-Common-Name From the ACLI

Use the following procedure to configure the P-Certificate-Subject-Common-Name.

To configure the P-Certificate-Subject-Common-Name:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type enforcement-profile, and press Enter.

```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```

4. add-certificate-info—Enter sub-common name for the certificate attribute names to enable TLS certificate information caching, and for the inserting of cached certificate information into customized SIP INVITEs. Default is blank. Valid values are:
   - sub-common name
   - sub-alt-name-DNS
5. certificate-ruri-check—Enable this parameter if you want your Oracle Enterprise Session Border Controller to cache TLS certificate information and use it to validate Request-URIs. Enabling this parameter allows the Net-Net ESD to cache the TLS certificate information in a customized SIP INVITE. Default is disabled. Valid values are:
   - enabled
   - disabled
6. verify-certificate-info-register —Select whether or not to allow the Oracle Enterprise Session Border Controller to add certificate information to the header of a REGISTER message for verifying a ruri against certificate attributes. Default is disabled. Valid values are:
   - enabled
   - disabled
7. Type done, and press Enter.

```
ACMEPACKET(enforcement-profile)# done
ACMEPACKET(enforcement-profile)#
```

8. Type exit, and press Enter.

```
ACMEPACKET(enforcement-profile)# exit
ACMEPACKET(session-router)#
```

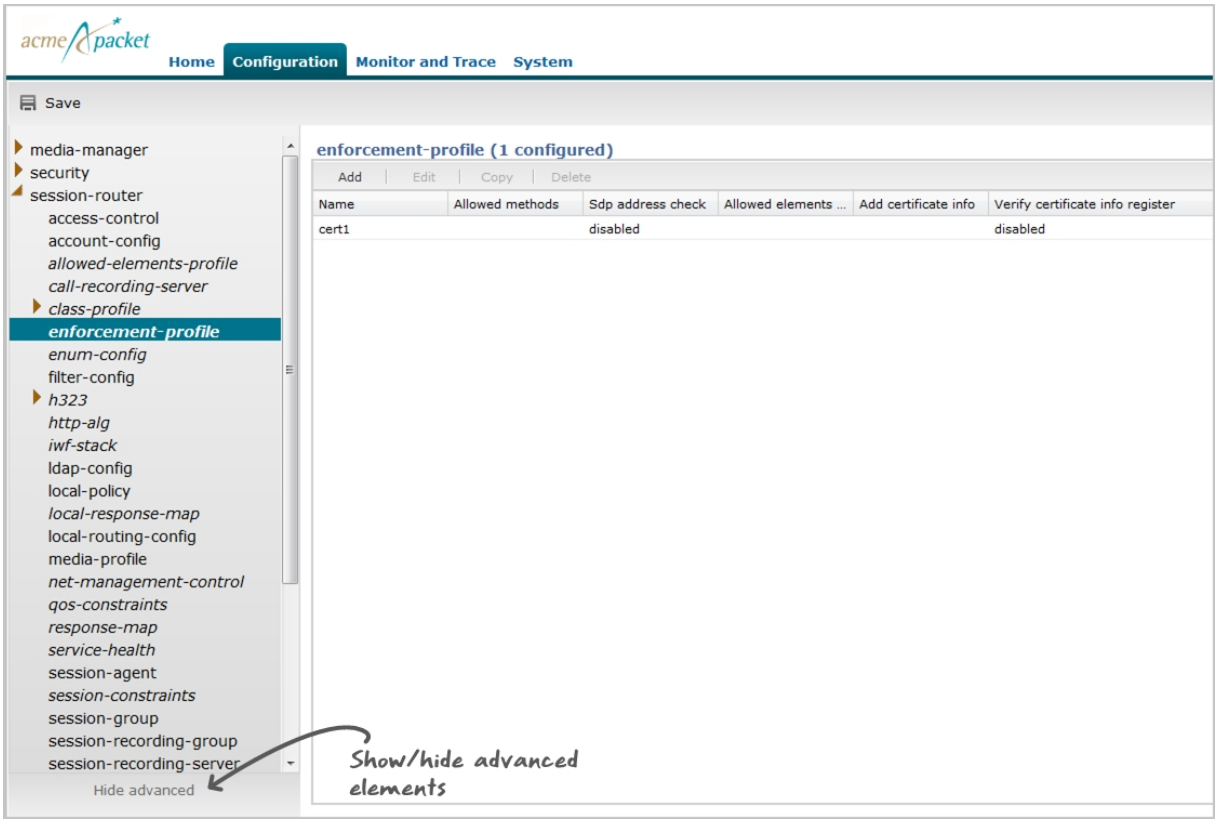
9. Save the configuration.

Configure the P-Certificate-Subject-Common-Name From the Web GUI

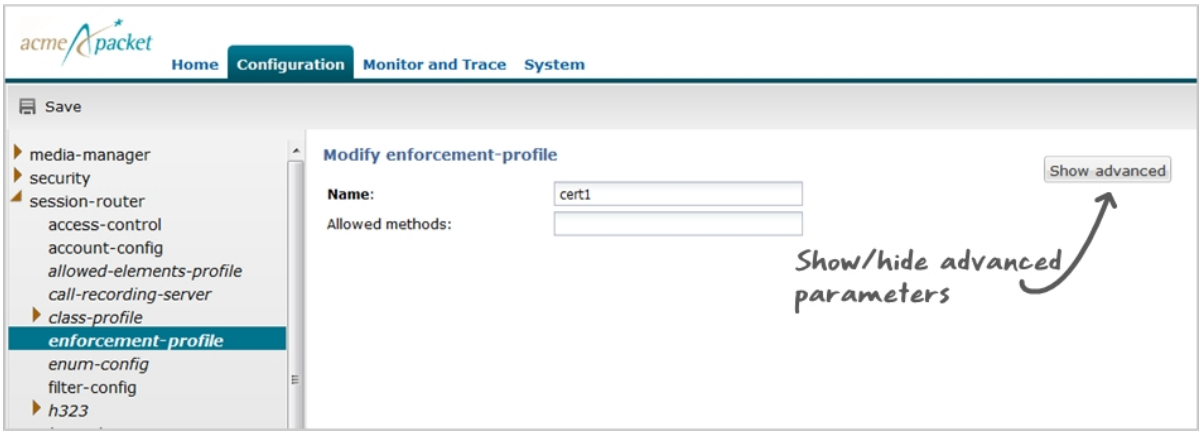
Use the following procedure to configure the P-Certificate-Subject-Common-Name using the Oracle Enterprise Session Border Controller Web GUI. In the Web GUI, this feature can be configured using Expert mode only and is an advanced configuration parameter.

To configure the P-Certificate-Subject-Common-Name in Expert mode:

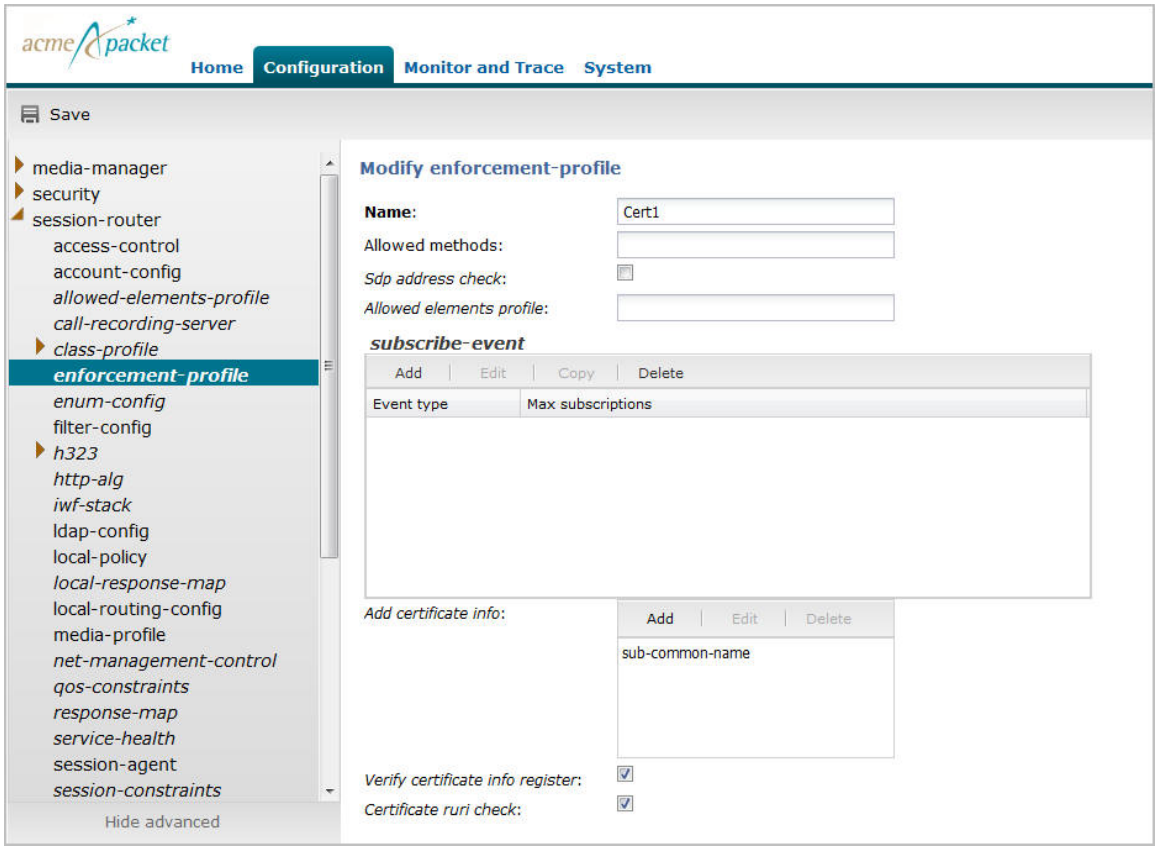
1. Logon to the Web GUI, and click Switch to Expert.
2. At the bottom of the left column, click Show advanced. The advanced elements for the objects in the left column display.



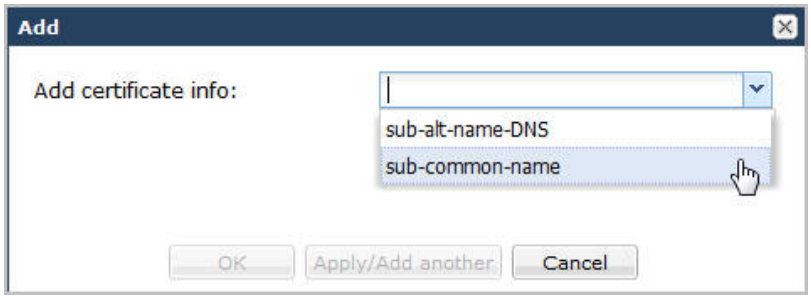
3. Click session-router.
4. Click enforcement-profile.
5. In the enforcement-profile dialog box, select the name of the certificate for which you want to enable the P-Certificate-Subject-Common-Name, and click <Edit>. The following dialog box displays.



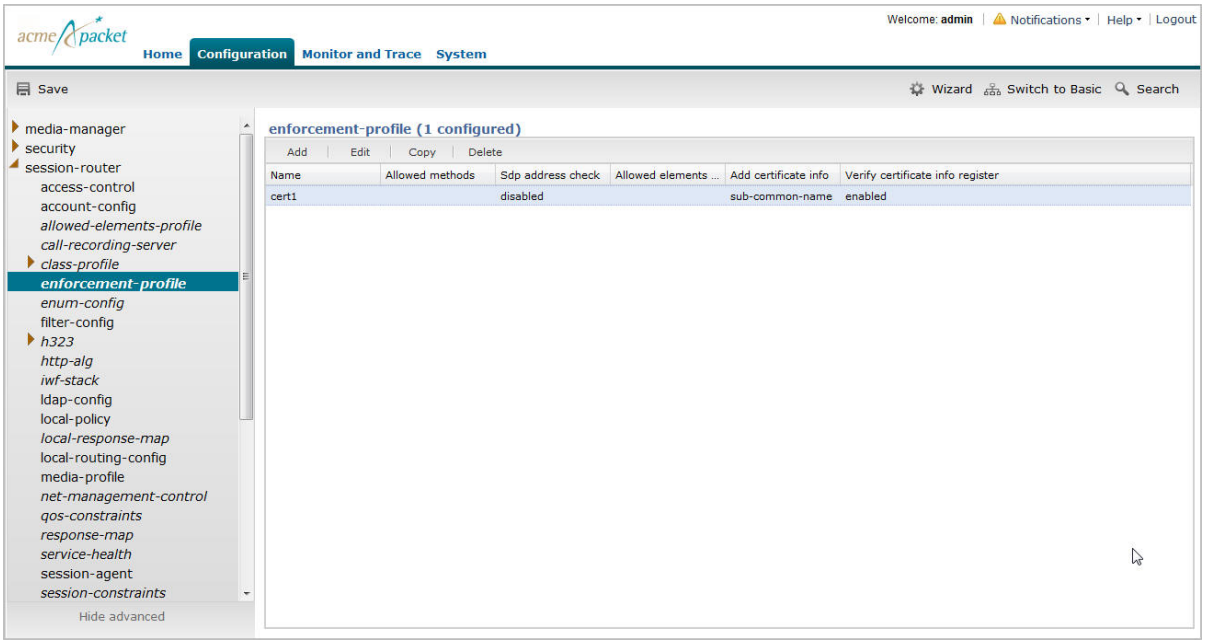
6. Click <Show advanced>. The advanced parameters for the certificate display.



7. In the Add certificate info box, click <Add>. The following dialog box displays.



8. Select sub-common-name from the drop-down box, and click <OK>.
9. In the Verify certificate info register field, place a check mark in the box to enable the Oracle Enterprise Session Border Controller to add certificate information to the header of a REGISTER message for verifying a ruri against certificate attributes. Click <OK>.
10. In the Certificate ruri check field, place a check mark in the box to enable this parameter if you want your Oracle Enterprise Session Border Controller to cache TLS certificate information and use it to validate Request-URIs. Enabling this parameter allows the Net- Net ESD to cache the TLS certificate information in a customized SIP INVITE. Click <OK>. The following window displays.



The certificate name has verify certificate info register enabled. The Oracle Enterprise Session Border Controller will include the sub-common name in the REGISTER message header before the UE registers.

Local Policy Session Agent Matching for SIP

When you enable the local policy session agent matching option in your global SIP configuration, you change the way local policies match session agents. Normally, the Oracle Enterprise Session Border Controller looks up and stores matched session agents configured as next hops so it does not need to perform the lookup while processing requests. In this type of matching, the Oracle Enterprise Session Border Controller does take the realm set in the local policy attributes into consideration. When the Oracle Enterprise Session Border Controller performs its regular matching method and you have enabled overlapping IP addresses for session agents, the Oracle Enterprise Session Border Controller might match session agents to different realms than the ones you intended when creating your configuration.

Local policy session agent matching provides a way to match session agents differently, taking realms and nested realms into consideration during the matching process. This difference is key to deployments with multiple peering partners that use the overlapping IP address feature, and have multiple local policies routing to the same IP address in different realms where some target next hops require session constraints but others do not. In the cases where no session constraints are required, session agents are not needed. But session agents still match the local policy, applying their constraints, because they match the next hop IP address.

In addition to modifying this behavior, this feature also affects the use of realms and nested realms. It triggers the use not only of realms, but of all the realms nested however deeply—thereby improving matching efficiency.

You can set the local policy session agent matching option with values that define how the Oracle Enterprise Session Border Controller performs session agent matching:

- any—The Oracle Enterprise Session Border Controller looks up and stores matched session agents configured as next hops so it does not need to perform the lookup while processing requests, without regard to realms.

  This behavior is the default when the SIP configuration does not have the local policy session agent matching option set.
- realm—The Oracle Enterprise Session Border Controllerselects session agents in the realm that the local policy attribute indicates; this provides an exact match, rather than not taking the realm into consideration during session agent selection.

For example, the session agent is a match if the session agent realm-id and the local policy attribute realm parameters are an exact match.

- sub-realm—Session agents in the same realm or the same realm lineage—where session agents and realms are related to one another via realm parent-child relationships no matter the depth of realm nesting configured

  For example, the session agent is a match if the local policy attribute realm is a sub-realm of the realm specified in the session agent realm-id parameter.
- interface—Session agents in the same realm or same realm lineage via the realm set in the local policy attribute, and whose realm uses the same signaling interface as the realm set in the local policy attribute

  For example, the session agent is a match if the session agent realm-id is a sub-realm of the local policy attribute realm, and both referenced realms use the same SIP signaling interface.
- network—Session agents whose realm is in the realm lineage for the same realm set in the local policy attributes, and whose realm is associated with the same network interface as the realm set in the local policy attributes

  For example, the session agent is a match if the session agent realm-id is a sub-realm of the local policy attribute realm, and realm reference by both use the same network interface.

If it cannot find a match, the Oracle Enterprise Session Border Controller will use the IP address as the next hop. Further, requests matching local policy attributes will not be associated with session agents, and so their constraints will not be applied.

The Oracle Enterprise Session Border Controller stores session agent information that it looks up when performing local policing session agent matching. To perform the lookup, it uses the session agent hostname as a key. When the hostname is an FQDN and there is a configured IP address in the ip-address parameter, the Oracle Enterprise Session Border Controller uses the ip-address value as a secondary key. Given this implementation, the following are true when selecting session agents:

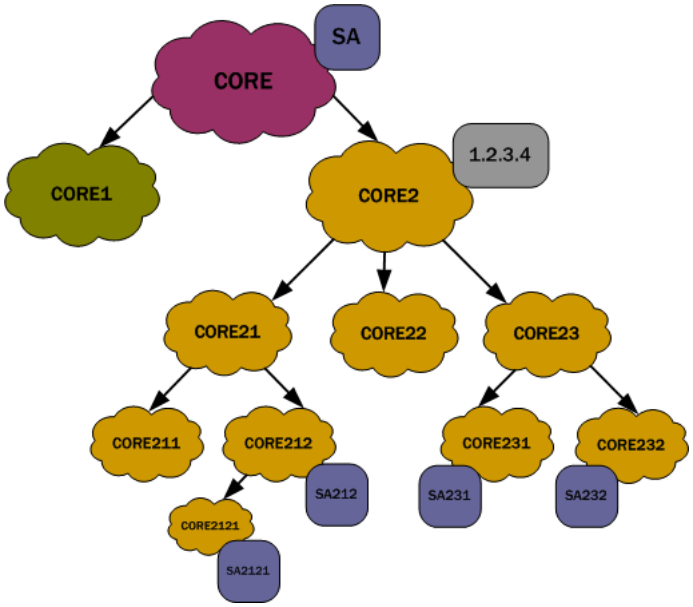
- If multiple session agents share the same IP address, the one with an IP address in the hostname parameter takes precedence.
- If all session agents with the same IP address have an FQDN as their hostname, the one whose name is alphabetically lower will take precedence, where alphabetically lower means earlier in the alphabet (closer to A than to Z).
- For non-global session agents (whose realms are configured but not wildcarded) with an IP address, the Oracle Enterprise Session Border Controller uses a key that is a combination of the IP address and the realm in the form <address>:<realm>.
- For a session agent whose realm has a parent realm, the Oracle Enterprise Session Border Controller uses a combination of the IP address, realm, and realm-path (or lineage for the realm) in the form <address>:<realm-path>. For example, the realm path for a realm core3 with a parent core2, which in turn has a parent core would be core:core2:core3.

When it looks up a session agent with a realm, the Oracle Enterprise Session Border Controller first searches for an exact match for the IP address and realm combination. If this fails, it performs a second search if the desired realm has parents or children. The Oracle Enterprise Session Border Controller locates an entry in its repository of session agent information that is greater than or equal to the IP address with the base realm, which is the ancestor of the desired realm without a parent. Having gathered this set of candidates, the Oracle Enterprise Session Border Controller narrows down the search for a match by comparing sub-realms and determines there is a match if either:

- The desired realm path is a sub-string of the entry's realm path, or
- The entry's realm path is a substring of the desired realm path (i.e., the desired realm is a sub-realm of the entry's realm)

Then the Oracle Enterprise Session Border Controller orders the candidates by depth of the entry's realm-path, or number of levels from the base realm relative to the depth of the desired realm. By searching the ordered set until the entry's realm depth equals the desired realm's depth, the Oracle Enterprise Session Border Controller determines a parent candidate, all subsequent entries being sub-realms of the desired realm. The Oracle Enterprise Session Border Controller only considers entries at the first level deeper than the desired realm. If at this point there is only one entry, the Oracle Enterprise Session Border Controller deems it a match. Otherwise, it selects the parent candidate as the matching entry. In the event the search does not yield a matching realm, the Oracle Enterprise Session Border Controller uses the global session agent for the IP address, if there is one.

The following diagram shows the realm tree, where the clouds are realms and squares are session agents, representing a group of session agents sharing the IP address 1.2.3.4. The Oracle Enterprise Session Border Controller searches for the session agents lower in the tree along the session agent realm-path and the desired realm.



For the diagram above, the following shows how the hostname would look for this group of session agents.

Key	Session Agent (hostname[realm])
1.2.3.4 (This session agent owns the primary key for the IP address because its hostname is the IP address.)	1.2.3.4[CORE2]
1.2.3.4:CORE (IP+realm key entry)	SA[CORE]
1.2.3.4:CORE (IP+realm key entry)	1.2.3.4[CORE2]
1.2.3.4:CORE212 (IP+realm key entry)	SA212[CORE212]
1.2.3.4:CORE2121 (IP+realm key entry)	SA2121[CORE2121]
1.2.3.4:CORE231 (IP+realm key entry)	SA231[CORE231]
1.2.3.4:CORE232 (IP+realm key entry)	SA232[CORE232]
1.2.3.4:CORE: (IP+realm-path key entry)	SA[CORE]
1.2.3.4:CORE:CORE2:	1.2.3.4[CORE2]

Key	Session Agent (hostname[realm])
(IP+realm-path key entry)	
1.2.3.4:CORE2:CORE21:CORE212 (IP+realm-path key entry)	SA212[CORE212]
1.2.3.4:CORE2:CORE21:CORE212:CORE2121 (IP+realm-path key entry)	SA2121[CORE2121]
1.2.3.4:CORE2:CORE23:CORE231 (IP+realm-path key entry)	SA231[CORE231]
1.2.3.4:CORE2:CORE23:CORE232 (IP+realm-path key entry)	SA232[CORE232]

For each realm in the table above, the search results for each realm would look like this:

IP Address	Realm	Session Agent (hostname[realm])
1.2.3.4	CORE	SA[CORE]
1.2.3.4	CORE2	1.2.3.4[CORE2]
1.2.3.4	CORE21	SA212[CORE212[
1.2.3.4	CORE211	1.2.3.4[CORE2]
1.2.3.4	CORE212	SA212[CORE212]
1.2.3.4	CORE2121	SA2121[CORE2121]
1.2.3.4	CORE22	1.2.3.4[CORE2]
1.2.3.4	CORE23	1.2.3.4[CORE2]
1.2.3.4	CORE231	SA231[CORE231]
1.2.3.4	CORE232	SA232[CORE232]

Local Policy Session Agent Matching Configuration

When you enable local policy session agent matching, remember that you can choose from five different ways to use the feature: all, realm, sub-realm, interface, and network.

This example shows you how to use the realm selection.

To enable local policy session agent matching using the realm method:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name lp-sa-match=X (where X is the local policy session agent matching method you want to use) with a plus sign in front of it. Then press Enter.

   Remember that if you do not specify a method, the system uses the all method.

   ```
   ACMEPACKET(sip-config)# options +lp-sa-match=realm
   ```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.
5. Save and activate your configuration.

- Unordered—Meaning that the endpoint can deliver data within regard for their stream sequence number

You set this preference in the network parameters configuration.

SCTP Delivery Mode Configuration

To set the SCTP delivery mode:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```
2. Type system and press Enter.

   ```
   ACMEPACKET(configure)# system
   ACMEPACKET(system)#
   ```
3. Type network-parameters and press Enter.

   ```
   ACMEPACKET(system)# network-parameters
   ACMEPACKET(network-parameters)#
   ```
4. sctp-send-mode—Leave this parameter set to its default (unordered) so data delivery can occur without regard to stream sequence numbering. If data delivery must follow stream sequence number, change this parameter to ordered.
5. Save and activate your configuration.

About Wildcarding

The Oracle Enterprise Session Border Controller supports wildcarding the event type in the subscribe-event configuration. To wildcard the value, you enter an asterisk (*) for the event-type parameter instead of typing in the name of an actual event type.

When you wildcard this value, the Oracle Enterprise Session Border Controller applies the subscription limitations you set across all event types. Or, if you have entered multiple subscribe-event configurations, the Oracle Enterprise Session Border Controller applies the wildcard limits across the event types for which you have not set limits.

Consider the following example of a configured enforcement profile with a wildcarded subscribe-event configuration:

```
enforcement-profile
        name                            rulefour
        allowed-methods
        sdp-address-check               disabled
        subscribe-event
                 event-type                     *
                 max-subscriptions              1
        subscribe-event
                 event-type                     xyz
                 max-subscriptions              0
        last-modified-by                admin@console
        last-modified-date              2008-11-11 12:49:27
```

In this example, the enforcement profile allows all subscriptions that are event type xyz for a user. But it allows only one maximum for every other subscription event type.

Monitoring

You can display the number of subscription dialogs per SUBSCRIBE event type using the ACLI show registration sipd subscriptions-by-user command. You can display this information per event type, or you can show data for all event types by wildcarding the event type argument.

STUN Server

The Oracle Enterprise Session Border Controller supports RFC 3489, which defines Simple Traversal User Datagram Protocol (UDP) through Network Address Translators (NATs). Known as STUN, this lightweight protocol that allows applications to:

- Discover the presence and types of both NATs and firewalls between themselves and the public Internet
- Determine the public IP addresses allocated to them by the NAT

SIP endpoints use the STUN protocol to find out the public IP addresses and ports for SIP signaling and RTP media transport. Then they can use the address and port information to create multimedia sessions with other endpoints on the public network.

You can define STUN servers functionality on a per-realm basis, allowing you set up multiple STUN servers.

About STUN Messaging

STUN messages uses six messages, three of which are used for Binding and three of which are uses for the Shared Secret. While it supports all three Binding messages (request, response, and error), the Oracle Enterprise Session Border Controller does not support the Shared Secret Request or the message integrity mechanism that relies on the shared secret. When acting as a STUN server, the Oracle Enterprise Session Border Controller responds to STUN binding requests in accordance with RFC 3489 and the rfc3489bis draft.

STUN messages can contain the following attributes:

Message Type	Attribute Description
MAPPED-ADDRESS	Appears in the Binding Response; contains the source IP address and port from which the Binding Request was sent to the STUN server.
XOR-MAPPED-ADDRESS	Appears in the Binding Response; contains the MAPPED-ADDRESS information encoded in a way the prevents intelligent NAT devices from modifying it as the response goes through the NAT.
SOURCE-ADDRESS	Appears in the Binding Response; contains the IP address and port from which the STUN server sent its response.
CHANGED-ADDRESS	Appears in the Binding Response; contains an alternate STUN server IP address and port, different from the primary STUN server port. The STUN client might use this attribute to perform the NAT tests described in RFC 3489.
CHANGE-REQUEST	Appears in the Binding Request; instructs the STUN server to send its response from a different IP address and/or port. The STUN client might use this attribute to perform the NAT tests described in RFC 3489.
RESPONSE-ADDRESS	Appears in the Binding Request; defines an IP address and port to which the STUN server should send its responses. Appears in the Binding Request;
REFLECTED-FROM	Appears in the Binding Response; reflects the IP address and port from which a Binding Request came. Only included when the Binding Request has used the RESPONSE-ADDRESS attribute.

Message Type	Attribute Description
UNKNOWN-ATTRIBUTES	Appears in the Binding Error; reflects the mandatory attributes in a Binding Request message that the server does not support.
ERROR-CODE	Appears in the Binding Error; indicates an error was detected in the Binding Request, and contains an error code and reason phrase.

To perform NAT discovery, the endpoint (STUN client) sends a Binding Request to the STUN server port (IP address and port) with which it is configured. The STUN server then returns either a;

- Binding Response—Allows the transaction to proceed
- Binding Error—Halts the transaction, and prompts the client to take the action appropriate to the response given in the ERROR-CODE attribute

When the transaction proceeds and the STUN server sends the Binding Response, that response contains the MAPPED-ADDRESS attribute, which contains the IP address and port from which the server received the request. The STUN client then uses the MAPPED-ADDRESS when sending signaling messages.

For example, a SIP endpoint sends Binding Requests from its SIP port to determine the public address it should place in SIP headers, like the Via and Contact, of the SIP requests it sends. When this SIP endpoint prepares to make or answer a call, it sends Binding Requests from its RTP port to find out the public address it should place in SDP included in an INVITE request or response.

STUN Server Functions on the Oracle Enterprise Session Border Controller

When the Oracle Enterprise Session Border Controller receives a STUN message, it first determines its message type. Only STUN Binding Requests are processed, and all other message types are dropped without response.

Then the Oracle Enterprise Session Border Controller examines the Binding Request's STUN attributes. It returns error responses if it finds any unsupported mandatory attributes. This takes the form of a Binding Error Response, containing the ERROR-CODE attribute with reason 420 (Unknown Attribute) and an UNKNOWN-ATTRIBUTES attribute with a list of the unsupported attributes. If the Oracle Enterprise Session Border Controller receives a Binding Request with attributes that do not belong in STUN Binding Requests, it returns the Binding Error Response with the ERROR-CODE attribute with reason 400 (Bad Request).

Next the Oracle Enterprise Session Border Controller determines whether to follow RFC 3489 procedures or rfc3489bis procedures. If the Transaction ID contains the STUN cookie, then the Oracle Enterprise Session Border Controller follows rfc3489bis procedures; if not, the it follows RFC 3489 procedures. Because it defines the procedures for testing the NAT to see what type of NAT it is, RFC 3489 procedures are most complex. Issues with reliability of those results have caused testing procedures and attributes to be deprecated in fc3489bis.

RFC 3489 Procedures

The Oracle Enterprise Session Border Controller (the STUN server) constructs the Binding Response and populates it with these attributes:

- MAPPED-ADDRESS and (optionally) XOR-MAPPED-ADDRESS—Containing the source IP address and port from which the server saw the request come
- SOURCE-ADDRESS—Containing the IP address and port from which the server will send the Binding Response
- CHANGED-ADDRESS—Containing the STUN server port that has a different address and different port from the ones on which the server request was received

If the Binding Request contains a RESPONSE-ADDRESS attribute, the server adds the REFLECTED-FROM attribute with the IP address and port from which the server saw the request come. Then the server sends the Binding Response to the IP address and port in the RESPONSE-ADDRESS attribute. If the RESPONSE-ADDRESS attribute's IP address and port are invalid, the server sends a Binding Error Response with an ERROR-CODE attribute reason 400 (Bad Request) to the client.

If the Binding Request contains a CHANGE-REQUEST attribute, the server sends Binding Response from the IP address and port matching the information in the CHANGE-REQUEST. The following variations can occur:

- If the IP address and port flags are set, the server selects the server port with a different IP address and different port.
- If only the IP address flag is set, the server selects the server port with a different IP address but with the same port.
- If only the port flag is set, the server selects the server port with the same IP address but with a different port.

The selected server port appears in the Binding Responses's SOURCE-ADDRESS attribute. When there is no CHANGE-REQUEST attribute, the server uses the server port on which the Binding Request was received.

Finally, the server encodes the outgoing message and sends it to the client at either:

- The destination IP address and port in the REPONSE-ADDRESS attribute, if it was present in the Binding Request.
- The MAPPED-ADDRESS.

rfc3489bis Procedures

If the Binding Request contains the appropriate cookie in its Transaction ID, the server constructs a Binding Response populated with the XOR-MAPPED-ADDRESS attribute. That attribute will contain the source IP address and port from which the server saw the request come. Then the server encodes and sends the message to the client from the IP address and port on which the request was received. The message is sent to the IP address and port from which the request came.

Monitoring

- STUN Server Statistics—You can display statistics for the STUN server using the ACLI show mbcd stun command when the STUN server has been enabled. However, if the STUN server has not been enabled since the last system reboot, the command does not appear and no statistics will be displayed.
- STUN Protocol Tracing—You can enable STUN protocol tracing two ways: by configuration or on demand.
  - By configuration—The Oracle Enterprise Session Border Controller's STUN protocol trace file is called stun.log, which is classified as a call trace. This means that when the system configuration's call-trace parameter is set to enabled, you will obtain STUN protocol information for the system. As with other call protocol traces, tracing data is controlled by the log-filter in the system configuration.
  - On demand—Using the ACLI notify mbcd log or notify mbcd debug commands, you enable protocol tracing for STUN. Using notify mbcd debug sets the STUN log level to TRACE. You can turn off tracing using the notify mbcd onlog or notify mbcd nodebug commands. Using notify mbcd nodebug returns the STUN log level back to its configured setting.

STUN Server Configuration

You configured STUN servers on a per-realm basis, one server per realm. To support that various NAT tests it describes, RFC 3489 requires that two different IP addresses and two different UDP port numbers be used for each server. So your STUN server will listen on a total of four STUN server ports. Although newer work does away with this requirement, the Oracle Enterprise Session Border ControllerC supports it for the purpose of backwards compatibility.

For each realm configuration with an enabled STUN server, untrusted ACL entries will be added to forward all packets received on the four STUN Server Port.

To enable STUN server support for a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. stun-enable—Set this parameter to enabled to turn STUN server support for this realm on. This parameter defaults to disabled, meaning STUN server support is off.
5. stun-server-ip—Enter the IP address for the primary STUN server port. The default for this parameter is 0.0.0.0.
6. stun-server-port—Enter the port to use with the stun-server-ip for primary STUN server port. The default is 3478.
7. stun-changed-ip—Enter the IP address for the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. This IP address must be different from than the one defined for the stun-server-ip parameter. The default for this parameter is 0.0.0.0.
8. stun-changed-port—Enter the port combination to define the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. The default for this parameter is 3479.
9. Save and activate your configuration.

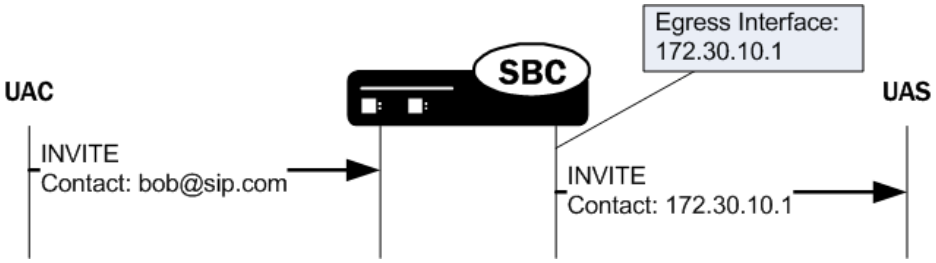
SIP GRUU

SIP Globally Routable User Agent (UA) URIs (GRUU) are designed to reliably route a SIP message to a specific device or end user. This contrasts with a SIP AoR which can refer to multiple UAs for a single user, thus contributing to routing confusion. The Oracle Enterprise Session Border Controller can perform different behaviors when it finds SIP GRUUs in Contact headers.

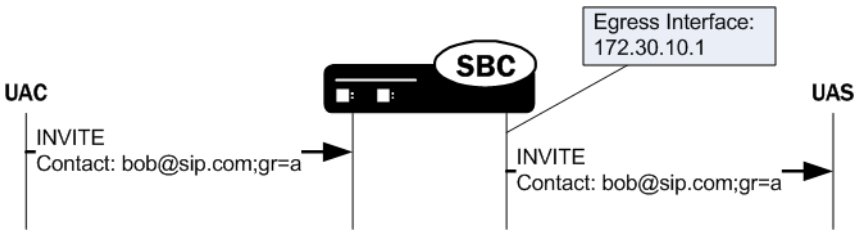
User agents supporting GRUU include a GRUU-identifying parameter in the Contact header of dialog forming and target refresh requests. The Oracle Enterprise Session Border Controller scans for the GRUU parameter in the Contact header either when the message is received on a realm or interface configured for registered endpoints or when the pass-gruu-contact parameter is enabled. Configure an interface for registered endpoints by enabling nat-traversal or registration caching.

Contact Header URI Replacement

When no GRUU is encountered in the contact header, and when a SIP message is forwarded to the egress realm, the contact header's URI is replaced with the Oracle Enterprise Session Border Controller's egress interface. For example:

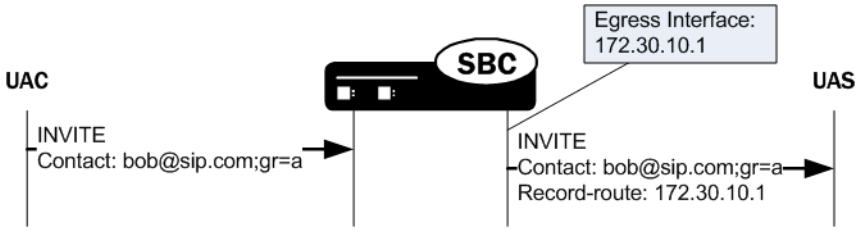


When the Oracle Enterprise Session Border Controller forwards a request where the original Contact header contains a GRUU, the contact header's URI is forwarded unchanged on the egress side of the call. For example:



Record-Route Addition

When the request is forwarded to a realm where the endpoint's registrar does not exist, the Oracle Enterprise Session Border Controller adds a Record-Route header containing the egress SIP interface address. This causes subsequent replies or requests addressed to the GRUU to be routed through the Oracle Enterprise Session Border Controller first.



When the request is forwarded to the realm where the registrar exists, adding the Record-Route header is unnecessary and does not occur. This is because subsequent requests are directed to the registrar which will ultimately forward them to the Oracle Enterprise Session Border Controller using the registered Contact in the Request-URI.

GRUU URI Parameter Name

The Oracle Enterprise Session Border Controller scans for a gr URI parameter in the contact header to identify it as a GRUU as defined in the ietf draft[2]. The Oracle Enterprise Session Border Controller can be configured to scan for a gruu URI parameter in the contact header too. This alternate behavior is enabled with the scan-for-ms-gruu option and is used to interact with the Microsoft Office Communications Server unified communications product. When scan-for-ms-gruu is enabled, the Oracle Enterprise Session Border Controller scans first for the gruu URI parameter. If not found, it then scans for gr URI parameter.

SIP GRUU Configuration

This section shows you how to configure the GRUU support for non-registered contacts. Enabling GRUU functionality to parse for gr URI parameter rather than the IETF standard gruu parameter is also provided.

To configure SIP GRUU functionality:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-config and press Enter.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI select command) before making your changes.

4. pass-gruu-contact—Set this parameter to enabled to parse for gr URI parameter in the contact header in non-registered endpoints' messages and then pass the messages thought the system.
5. options—Set the options parameter by typing options, a Space, the option name scan-for-ms-gruu. This option forces the Oracle Enterprise Session Border Controller to first scan for the gruu URI parameter, then the gr URI parameter.
6. Save and activate your configuration.

SIP Session Timer Feature

SIP does not have a keepalive mechanism for established sessions and it does not have the capability of determining whether a session is still active. User agents (UAs) may be able to determine whether a session has timed out by using session specific mechanisms, but proxies cannot always determine when sessions are still active.

The Oracle Enterprise Session Border Controller provides a SIP session timer feature that, when enabled, forwards the re-INVITE or UPDATE requests from a User Agent Client (UAC) to a User Agent Server (UAS) in order to determine whether or not a session is still active. This refresh feature works for both UAs and proxies. The following paragraphs provide additional information about the session timers on the Oracle Enterprise Session Border Controller.

How the Session Timer Feature Works

During an active SIP call session, when a UA fails to send a BYE message at the end of the session, or when the BYE message gets lost due to network problems, the proxy cannot determine when the session has ended. Therefore, the proxy may hold onto resources associated with the call session for indefinite periods of time causing the session to never time out.

The SIP session timer feature adds the capability to periodically refresh SIP sessions by sending repeated INVITE (re-INVITE) or UPDATE Session Refresh Requests. These requests are sent during active call legs to allow UAs or proxies to determine the status of a SIP session. The Session Refresh Requests along with the session timers determine if the active sessions stay active and completed sessions are terminated.

When you enable the session timer feature on the Oracle Enterprise Session Border Controller, it periodically sends out a Session Refresh Request (re-INVITE or UPDATE). The Response that is returned to the Oracle Enterprise Session Border Controller contains a success status code (2xx) that contains a session timer interval. The Oracle Enterprise Session Border Controller then refreshes the session timer each time it receives the 2xx Response containing that session timer interval.

The initial INVITE message sent from the UAC to the UAS contains two fields that make up the session timer interval in the SIP Session Header:

- Session-Expires (SE) - Specifies the maximum amount of time, in seconds, that can occur between session refresh requests in a dialog before the session is considered timed out.
- Minimum-SE (Min-SE) - Specifies the minimum allowed value, in seconds, for the session expiration.

The following displays the session timer interval values inserted in the SIP session INVITE message per RFC 4028:

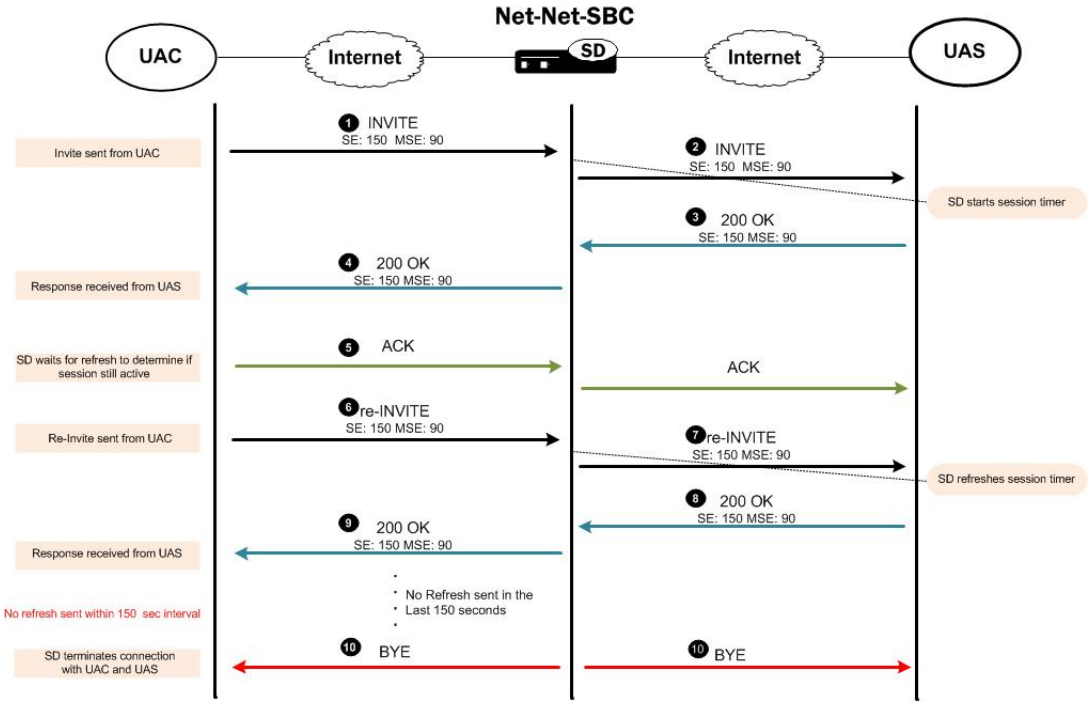
```
INVITE sip:9109621001@192.168.200.99 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.49:5060;branch=z9hG4bK0g6t23200gd0res4l580.1
Max-Forwards: 69
From: <sip:rick@192.168.1.48>;tag=SDr08od01-188c3fbc-b01a-4d68-
b741-09e5dc98a064
To: sip:149@192.168.1.49
Contact: <sip:rick@192.168.200.49:5060;transport=udp>
Call-ID: SDr08od01-9c12b48e3b0f7fad39ff3a2e0ced5ed3-v3000i1
CSeq: 3941 INVITE
Allow: INVITE, ACK, BYE, CANCEL, UPDATE, PRACK
Supported:  timer
Session-Expires: 1800
Min-SE: 90
Content-Type: application/sdp
Content-Length: 236

v=0
o=- 3462189550 3462189550 IN IP4 192.168.200.49
s=pjmedia
c=IN IP4 192.168.200.49
t=0 0
m=audio 20000 RTP/AVP 0 8 101
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=sendrecv
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

If the Oracle Enterprise Session Border Controller receives an INVITE from the UAC with a Session-Expires header, it starts a new session timer, or refreshes an existing session timer and then forwards the INVITE to the UAS. The subsequent 2xx Responses and re-INVITES also include the session timer intervals. If the Oracle Enterprise Session Border Controller does not receive a session refresh within the time specified in the session timer interval, the session timer expires, and the Oracle Enterprise Session Border Controller terminates the session between the UAC and the UAS.

The following occurs when you enable the session timer feature on the Oracle Enterprise Session Border Controller:

1. The UAC sends an INVITE to the Oracle Enterprise Session Border Controller with the SE and min-SE values (session timer interval). The Oracle Enterprise Session Border Controller starts a session timer.
2. The Oracle Enterprise Session Border Controller forwards the INVITE to the User Agent Server (UAS) with the same values.
3. The UAS sends the 200 OK Response to the Oracle Enterprise Session Border Controller with the session interval values.
4. The Oracle Enterprise Session Border Controller forwards the Response to the UAC.
5. The UAC sends an ACK (Acknowledge) to theOracle Enterprise Session Border Controller, and the Oracle Enterprise Session Border Controller forwards the ACK to the UAS.
6. The UAC sends out a re-INVITE (Session Refresh Request) to the Oracle Enterprise Session Border Controller with the session interval values. The Oracle Enterprise Session Border Controller refreshes the session timer.
7. The Oracle Enterprise Session Border Controller forwards the re-INVITE to the UAS.
8. The UAS sends the 200 OK response to the Oracle Enterprise Session Border Controller with the session interval values.
9. The Oracle Enterprise Session Border Controller sends the 200 OK response to the UAC.
10. If the Oracle Enterprise Session Border Controller does not receive a Response within the session interval (150 seconds in the following illustration), the timer expires, and the Oracle Enterprise Session Border Controller terminates the session between the UAC and the UAS. The following illustration shows an example of a dialog between the UAC, the Oracle Enterprise Session Border Controller, and the UAS during an active session.



When the Oracle Enterprise Session Border Controller terminates a session it sends a BYE to both the ingress and egress call legs. If accounting is configured, the Oracle Enterprise Session Border Controller also sends a RADIUS stop record with Acct-Terminate-Cause = Session-Timeout. You can enable or disable the use of the session timers using the ACLI interface at session-router > sip-config > options.

SIP Session Timer Configuration

You can configure the session timer feature on the Oracle Enterprise Session Border Controller to periodically refresh SIP sessions and determine whether or not a session is still active. If the Oracle Enterprise Session Border Controller determines that a session is no longer active, it terminates the session based on the session timer interval settings.

To configure the session timer feature on the Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session router-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter to access the SIP config-related configurations. The system prompt changes to let you know that you can begin configuring individual parameters for this object.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

4. Enter options followed by the following value:

   - +session-timer-support

     ```
     ACMEPACKET(sip-config)# options +session-timer-support
     ```

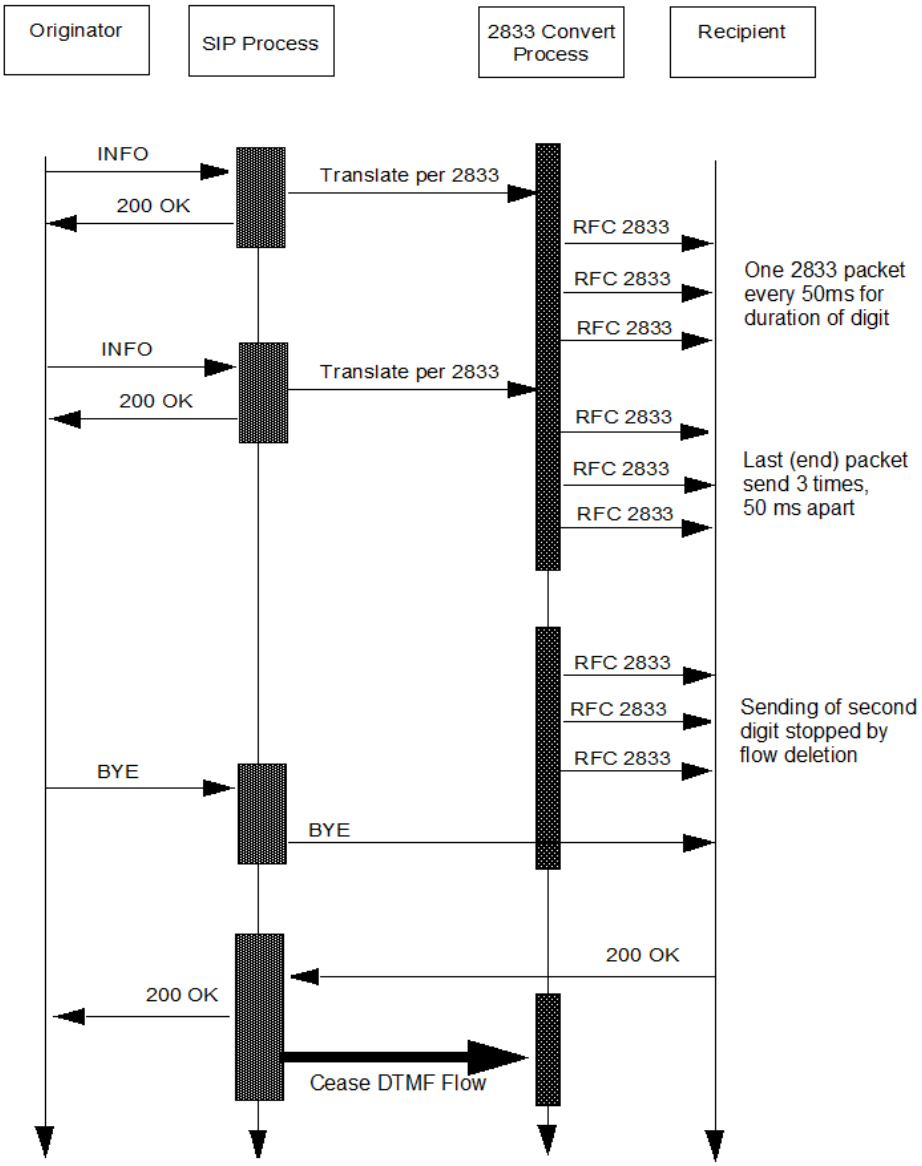
     This value enables the system to start the session timer for session refreshes coming from the UAC. The system determines whether or not a session is active based on session refreshes or responses. It terminates the session when no session refreshes occur within the session timer interval. To disable the session timer feature, enter options followed by -session-timer-support.

     ```
     Oracle Enterprise Session Border Controller(sip-config)# options -session-
     timer-support
     ```

Note: To disable session timers, you must add a minus sign (-) before the session-timers-support value.

DTMF Conversion Processing

Release S-CX6.3F1 provides a configurable SIP option to implement DTMF-to-RFC2833 tone translation. The current, default implementation, which performs well in most network topologies is shown below.



When the Oracle Oracle Enterprise Session Border Controller receives an INFO DTMF request, the SIP process determines whether or not it needs to perform DTMF-to-RFC2833 translation. If translation is required, the process forwards the DTMF to the 2833 convert process for translation and transmission to the recipient. Immediately after off-loading the DTMF, the SIP process sends a 200 OK response for the INFO. As shown in the figure, the 2833 convert process generates a number of RFC2833 packets to represent received DTMF digits.

Specifically, the 2833 convert process generates one RFC 2833 packet every 50 milliseconds for the duration of the DTMF digit, whose length is specified in the INFO request, and two retransmits of the last packet (known as the end packet) 50 milliseconds apart.

Consequently, the time interval between the 200 OK and the actual transmission of the RFC 2833 translation is the sum of the DTMF duration and 100 ms.

Note: This time interval can be shortened to 100 ms by enabling the rfc2833-end-pkts-only-for-non-sig parameter in media-manger which results in SD only generating the last packet and its two retransmits.

The early 200 OK allows the endpoint to send the next DTMF digit before the SD has sent all the RFC2833 packets, resulting in the next digit being queued internally by the 2833 convert process before being sent.

A problem arises if the SIP process receives a BYE request from the DTMF originator while queued digits are awaiting translation and transmission. In such an event, the SIP process immediately forwards the BYE request to the recipient, ending the session with DTMF digits awaiting translation and transmission.

An alternative DTMF conversion model provides for a feedback mechanism from the 2833 convert process to the SIP process. With this model enabled, the SIP process buffers a received BYE until it obtains confirmation that all queued DTMF digits have been translated and transmitted. Only after obtaining confirmation, does it forward the BYE to terminate the session.

This processing model is enable by a SIP option, sync-bye-and-2833, and requires that rfc2833-mode parameter on the egress interfaces is NOT set to dual , any value other than dual , is supported.

1. From superuser mode, use the following command sequence to access sip-config configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

2. Use the SIP option sync-bye-and-2833 to delay BYE processing until DTMF-to-RFC2833 translation has been completed.

```
ACMEPACKET(sip-config)# options +sync-bye-and-2833=”enabled”
ACMEPACKET(sip-config)#
```

3. Use the done and exit commands to complete configuration.

```
ACMEPACKET(sip-config)# done
ACMEPACKET(sip-config)# exit
ACMEPACKET(session-router)#
```

LMSD SIP Call Progress Tone Interworking

The Oracle Enterprise Session Border Controller supports Legacy Mobile Station Domain interworking tht allows SIP interworking with User Agents that support the 3GPP2 LMSD. The LMSD provides support for existing Mobile Stations in a network that supports IP bearers.

LMSD uses Alert-Info headers in a 180 Ringing response to an INVITE to indicate to a User Agent specific tones to generate locally by the UAC. Most User Agents that do not support LMSD will ignore the SDP in the Alert-Info and will not play ringback locally. The lmsd-interworking option allows for the system to suppress SDP in 180 Ringing, 486 Busy Here, and 503 Service Unavailable responses, so that the UAC plays local ringback.

There are no additional license requirements.

LMSD Interworking Configuration

You can apply lmsd-interworking to the sip-interface facing the LMSD User Agents. For Example, if the endpoints uspporting LMSD are located in the core realm, then the lmsd-interworking option would be added to the core sip-interface.

To enable the LMSD Interworking option:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. options—Set the options parameter by typing options, a Space, the option name lmsd-interworking with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-interface)# options +lmsd-interworking
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save your work.

SIP re-INVITE Suppression

Some of the Interactive Voice Response (IVR) systems that support SIP frequently change the media transport address (IP address and/or port number) when switching between voice menus and/or prompts by sending a re-INVITE.

Often, no other parameters or properties of the session are changed in these re-INVITEs. The frequent re-INVITEs can create performance and capacity problems in other systems along the path of the IVR system and the caller's User Agent.

You can configure the suppress-reinvite option on your Oracle Enterprise Session Border Controller, allowing it to store the previous INVITE and its 200-OK response. Having this information allows the system to reply locally when a re-INVITE that changes only the media transport addresses is recieved.

No license requirements to enable this feature.

SIP re-INVITE Suppression Configuration

To enable SIP re-INVITE Suppression:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. options—Set the options parameter by typing options, a Space, the option-name suppress-reinvite with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(sip-interface)# options +suppress-reinvite
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the SIP interface configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

RFC 4028 Session Timers

The Oracle Enterprise Session Border Controller supports RFC 4028 Session Timers. In this role, it acts as a B2BUA between two endpoints and then enforces the timer values on each call leg independently. The RFC 4028 abstract states:

This document defines an extension to the Session Initiation Protocol (SIP). This extension allows for a periodic refresh of SIP sessions through a re-INVITE or UPDATE request. The refresh allows both user agents and proxies to determine whether the SIP session is still active. The extension defines two new header fields:

- Session-Expires—which conveys the lifetime of the session
- Min-SE—which conveys the minimum allowed value for the session timer

The following parameters in the session-timer-profile configuration element are used for this feature:

session-expires—The value of the session expires header in seconds

min-se—The value of the Min-SE header in seconds (this is a minimum session expires value)

force-reinvite—Sets if the Oracle Enterprise Session Border Controller will send a reINVITE to refresh the session timer when applicable.

request-refresher—Set on the outbound side of a call what the Oracle Enterprise Session Border Controller sets the refresher parameter to. Valid values are uac, uas, or none.

response-refresher—Set on the inbound side the value of the refresher parameter in the 200OK message. Valid values are uac or uas.

In this section, the notion that a UAC or UAS supports Session Timers is indicated by the presence of the Supported: timer header and option tag.

Ingress Call Leg

Setting 200 OK's Session-Expire value

The session timer is based on the negotiation between each side's session expires value. The final value on the ingress leg is returned by the Oracle Enterprise Session Border Controller to the UAC, unless there is an error.

The Oracle Enterprise Session Border Controller can always reduce the session expires value it returns to the UAC. It checks that the Session-Expires: header is larger than the SIP Interface's min-se value. The Oracle Enterprise Session Border Controller then compares the received Session-Expires: header to the configured session-expires configuration element and uses the lower value for the 200 OK's Session-Expires: header. If this outbound Session-Expires: value is lower than the received Min-SE: header, it will be bumped up to the Min-SE: header's value.

If the Oracle Enterprise Session Border Controller's Min-SE value is larger than the Session-Expires: header, a 422 (Session Interval Too Small) message is returned to the UAC containing the Oracle Enterprise Session Border Controller's configured Min-SE value.

When the UAC supports (but does not require) Session Timers and the Oracle Enterprise Session Border Controller does not support session timers, a 200 OK is returned to the UAC with no indication of session timer support.

Refresher

The initial UAC, the side that sends the INVITE, can set itself to be the refresher (uac) or the Oracle Enterprise Session Border Controller as the refresher (uas). Whoever is the refresher is indicated in the 200 OK. If the UAC does not specify any refresher, the Oracle Enterprise Session Border Controller uses it's response-refresher value in the 200 OK. If that value is set to uas, the Oracle Enterprise Session Border Controller creates and sends a re-INVITE toward the UAC with previously negotiated session expiration values.

Once the Oracle Enterprise Session Border Controller becomes the refresher, it does not relinquish that role. Then, when the Oracle Enterprise Session Border Controller sends refresh requests, it does not change any parameters (refresher role & timers) from the initial request negotiation.

UAC does not Support Session Timers

If the UAC's initial request does not include a Session-Expires: header, then the 200 OK will include the session-timer-profile > session-expires value on the ingress leg in the Session-Expires: header.

The Oracle Enterprise Session Border Controller also inserts the refresher parameter as configured. The orientation of UAC/UAS on the Oracle Enterprise Session Border Controller's view of a call leg can change if later in the call flow the endpoint designates the Oracle Enterprise Session Border Controller as the refresher.

Note: When the request doesn't support Session Timers, the Oracle Enterprise Session Border Controller's reply adds session timer support according to configuration.

If the Oracle Enterprise Session Border Controller receives a message with a Require: timer header, and the inbound SIP interface or the final UAS do not support session timers, a 420 (Bed Extension) is returned to the UAC.

Egress Call Leg

Outbound INVITE Message

When the Oracle Enterprise Session Border Controller's outbound interface is configured with session timers, it forwards an INVITE to the UAS with the following headers:

Session-Expires— Oracle Enterprise Session Border Controller inserts the outbound SIP interface's session-timer parameter

Session-Expires refresher parameter— Oracle Enterprise Session Border Controller inserts the request-refresh parameter

Min-SE— Oracle Enterprise Session Border Controller inserts the outbound SIP interface's session-timer parameter

Supported—Supported header has the timer option tag

Note: Require/Proxy-Require—If the timer parameter is present in the Require or Proxy-Require: header field in the request received from the UAC, it will be removed.

No Session Timer Configuration

If the ingress SIP interface supports session timers, and the original INVITE from the UAC included session timer support, the INVITE request sent to the UAS will have no session timer support. However, the Supported: timer header will be created. This ensures that the Oracle Enterprise Session Border Controller does not get a 421 response for 'timer' from the UAS.

If the ingress SIP interface supports session timers, and the UAC's initial INVITE did not include session timer support, then the INVITE sent to the UAS will have no session timer support (headers) as well.

If the ingress SIP interface does not support session timers, the INVITE is forwarded with no Session Timer alteration.

UAS Initial Response

Upon receiving a 200 OK from the UAS, if the response specifies uac as the refresher, the 200 OK includes a Session-Expires header and specifies uac as the refresher, the Oracle Enterprise Session Border Controller will assume the refresher role. If the 200 OK does not include a Session-Expires header, and the egress interface supports session timers, then the Oracle Enterprise Session Border Controller assumes the refresher role.

UAS Returns Errors

422 Session Interval Too Small—The Oracle Enterprise Session Border Controller in response sends the request again with new values in the 'Session-Expires' header field based on the 'Min-SE' value present in the 422 response.

421 Extension Required for 'timer'—This response can only happen if none of the other three entities (UAC, ingress SIP interface and egress SIP interface) support session timers. The 421 is forwarded through the system to the original UAC.

420 Bad Extension for 'timer'—This response should never happen because the Oracle Enterprise Session Border Controller will never send Require: timer header. But the event this error is received, it will be forwarded to the original UAC.

Session Refreshes

On either side of the call, the Oracle Enterprise Session Border Controller can be responsible for initiating the session refreshes or responding to the session refreshes.

Oracle Enterprise Session Border Controller as Refresher

The Oracle Enterprise Session Border Controller sends the refresh request when half the session expiration has elapsed. The Oracle Enterprise Session Border Controller always wants to remain the refresher and maintain the initially agreed upon session expiration timers.

Creating the Refresh Message

The refresh message takes the form of a re-INVITE when the force-reinvite parameter in the session timer profile is enabled. If this parameter is disabled and the remote end supports UPDATE requests, an UPDATE message will be sent.

UPDATE messages contain no SDP information.

Re-INVITE messages contain the SDP that is the same as what was sent before.

The refresh request's Session-Expires: header value is set to the existing value for the session. The refresher parameter is set to uac since the Oracle Enterprise Session Border Controller acts like a UAC for this refresh transaction. The Min-SE header is also included.

Processing the Refresh Response

The session expires value in the 2xx response is accepted and the timer restarts.

If the remote end does not include any session expiration parameters, the Oracle Enterprise Session Border Controller continues to support session timers, and assumes that the refresh interval is the same as before.

Any response that is 422 Session Interval Too Small is handled as expected. The Oracle Enterprise Session Border Controller resends the refresh request again with new values based on the 422 response. Any other response to the refresh request that is not a dialog/usage destroying response is treated like a 200 OK response.

Subsequent refresh requests are created and sent after half the previous refresh interval. If non-2xx, dialog / usage destroying responses are received, the Oracle Enterprise Session Border Controller reduces the following refresh intervals by half, as long as the final interval is not less than 32 seconds. The Oracle Enterprise Session Border Controller then uses this period for sending refresh requests until it successfully receives a 2xx response.

Oracle Enterprise Session Border Controller as Refresh Responder Processing the Refresh

The refresh request is processed similarly to the initial request regarding the session timer parameters. The session timer for this call leg is restarted when the Oracle Enterprise Session Border Controller when it sends the 200 OK response for the refresh request.

Forwarding the Refresh

When the Oracle Enterprise Session Border Controller receives an UPDATE request, it is forwarded to the other end since the Oracle Enterprise Session Border Controller cannot determine whether this request is only for session refreshing, or for other purposes as well.

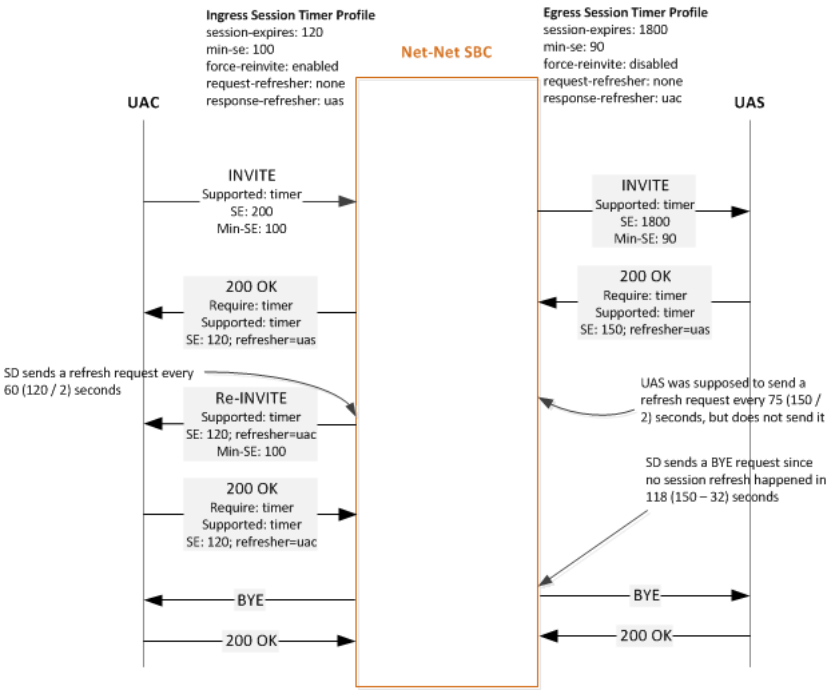
When the Oracle Enterprise Session Border Controller receives a re-INVITE request, it will determine whether this request needs to be suppressed, or should it be forwarded to the other end.

Timer Expiration

If the Oracle Enterprise Session Border Controller fails to receive a session refresh request before the session expiration, the session will be terminated before the full session time. This is computed according to:

$$time = period - min(\frac{period}{3}, 32)$$

After the real expiration time elapses, the Oracle Enterprise Session Border Controller sends a BYE request in both directions to terminate the session.



Interaction with SIP Features

Consider the following sections that have interactions with RFC 4028 Support.

sip-config option session-timer-support

A configured session-timers-profile on a SIP interface overrides the session-timer-support option in the SIP config. The Oracle Enterprise Session Border Controller can still act in proxy mode for some calls and B2BUA for other calls considering which SIP interfaces session timer profiles are not configured for.

sip-feature Support

When the UAC sends a Require: timer header is in the initial request and the Oracle Enterprise Session Border Controller does not support session timers, and no sip-feature configuration element is configured for 'timer' for that realm, the Oracle Enterprise Session Border Controller replies with a 420 (Bad Extension) response for 'timer'.

When the UAC sends a Require: timer header is in the initial request and the Oracle Enterprise Session Border Controller does support session timers, the timer tag is removed from the Require: header even if a sip-feature configuration element is configured for 'timer'. This also applies for the Proxy-Require: header.

Oracle recommends you do not configure a sip feature configuration element while using the Session Timers feature.

sip-interface option suppress-reinvite

SIP re-INVITE suppression is automatically enabled for a SIP interface when session timers are enabled. This behavior prevents re-INVITEs whose purpose is only for session refreshes from being forwarded to the other call leg. The first re-INVITE received from a UAS on the terminating call leg will be passed to the UAC on the originating call leg since the Oracle Enterprise Session Border Controller has no prior INVITE request coming from the UAS to match against.

Re-INVITEs are suppressed only when the Oracle Enterprise Session Border Controller receives the same INVITE request back-to-back without any intervening re-INVITE request in the opposite direction, or an UPDATE or PRACK request in either direction.

Note: The SIP reINVITE Supression parameters are not replicated on the standby system in an HA environment. The first re-INVITE after a switchover will be forwarded to the far end.

Examples

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
1	INVITE → Supported: timer SE: 200 ←— 200 OK Require: timer SE: 200; refresher=uas	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas this element becomes refresher	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas	INVITE→ Supported: timer SE: 500 Min-se: 400 ←— 200 OK Require: timer SE: 400; refresher=uas
2	INVITE → Supported: timer SE: 1200; refresher=uas ←— 200 OK Require: timer SE: 500; refresher=uas	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uac this element becomes refresher	session-expires: 500 min-se: 400 request-refresher: uas response-refresher: uas this element becomes refresher	INVITE → Supported: timer SE: 500; refresher=uas Min-se: 400 ←— 200 OK
3	INVITE → Supported: timer SE: 1200; refresher=uac Min-se: 800 ←— 200 OK Require: timer SE: 800; refresher=uac	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas	No session timer configuration this element becomes refresher	INVITE → Supported: timer ←— 200 OK Require: timer SE: 400; refresher=uac
4	INVITE → Supported: timer SE: 200; refresher=uac ←—	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas	No session timer configuration	INVITE → Supported: timer ←— 200 OK

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	← 200 OK Require: timer SE: 200; refresher=uac			
5	INVITE → Supported: timer SE: 200 ——————————— ←— 200 OK	No session timer configuration	session-expires: 500 min-se: 400 request-refresher: uas response-refresher: uas this element becomes refresher	INVITE → Supported: timer SE: 500; refresher=uas Min-se: 400 ——————————— ←— 200 OK
6	INVITE → Supported: timer SE: 200 ——————————— ←— 200 OK Require: timer SE: 400; refresher=uas	No session timer configuration SBC behavior stays same as current behavior	No session timer configuration	INVITE → Supported: timer SE: 200 ——————————— ←— 200 OK Require: timer SE: 400; refresher=uas
7	INVITE → Require: timer SE: 200 ——————————— ←— 420 Unsupported: timer	No SIP feature for timer No session timer configuration SD behavior stays same as current behavior	No session timer configuration	
8	INVITE → Require: timer SE: 200 ——————————— ←— 420 Unsupported: timer	SIP feature configured for timer No session timer configuration SD behavior stays same as current behavior	No session timer configuration	INVITE → Required: timer SE: 200 ——————————— ←— 420 Unsupported: timer
9	INVITE → Require: timer SE: 200	SIP feature configured for timer No session timer configuration	session-expires: 500 min-se: 400 request-refresher: none	INVITE → Supported: timer SE: 500

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	—————————— —— ← 200 OK		response-refresher: uas this element becomes refresher	Min-se: 400 —————————— —— ← 200 OK Require: timer SE: 500; refresher=uac
10	INVITE → Require: timer SE: 200 —————————— —— ← 200 OK Require: timer SE: 200; refresher=uac	No SIP feature for timer session-expires: 500 min-se: 200 request-refresher: none response-refresher: uac	session-expires: 500 min-se: 400 request-refresher: uas response-refresher: uas	INVITE → Supported: timer SE: 500; refresher=uas Min-se: 400 —————————— —— ← 200 OK Require: timer SE: 500; refresher=uas
11	INVITE → Require: timer SE: 200 —————————— —— ← 420 Unsupported: timer	No SIP feature for timer No session timer configuration	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas	
12	INVITE → SE: 200 —————————— —— ← 200 OK SE: 500; refresher=uas	session-expires: 500 min-se: 500 request-refresher: none response-refresher: uas this element becomes refresher	No session timer configuration	INVITE → —————————— —— ← 200 OK
13	INVITE → —————————— —— ← 200 OK	No session timer configuration	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas	INVITE → Supported: timer SE: 500 Min-se: 400 —————————— —— ← 200 OK Require: timer

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
				SE: 400; refresher=uas
14	INVITE → ——————————— ——— ← 421 Require: timer	No session timer configuration SD behavior stays same as current behavior	No session timer configuration	
15	INVITE → Supported: timer SE: 200 ——————————— ——— ← 422 Min-se: 400	session-expires: 500 min-se: 400		
16	INVITE → Supported: timer SE: 200 ——————————— ——— ← 200 OK Require: timer SE: 200; refresher=uac	session-expires: 800 min-se: 90 request-refresher: none response-refresher: uac	session-expires: 800 min-se: 90 request-refresher: none response-refresher: uac	INVITE → Supported: timer SE: 800 Min-se: 90 ——————————— ——— ← 422 Min-se: 900 ——————————— ——— INVITE → Supported: timer SE: 900 Min-se: 900 ——————————— ——— ← 200 OK Require: timer SE: 900; refresher=uas

RADIUS Interim record Generation

When refresh requests (UPDATE or Re-INVITE) are sent by the Oracle Enterprise Session Border Controller , no RADIUS Interim records are generated because session parameters do not change when these requests are sent.

When UPDATE requests are received by the Oracle Enterprise Session Border Controller , no RADIUS Interim records are generated.

When Re-INVITE requests are received by the Oracle Enterprise Session Border Controller , RADIUS Interim records are generated if the generate-interim parameter is enabled.

ACLI Configuration

To configure a session timer profile object:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-timer-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-timer-profile
ACMEPACKET(session-timer-profile)#
```

4. name—Enter a name for this session timer profile.
5. session-expires—Enter the session timer value in seconds you wish this object to use natively.
6. min-se—Enter the minimum session timer value in seconds for this object.
7. force-reinvite—Leave the default of enabled for the Oracle Enterprise Session Border Controller to always use reINVITEs for session refreshes. Set this parameter to disabled for the Oracle Enterprise Session Border Controller to try using UPDATEs for session refreshes.
8. request-refresher—Set this to the value to insert in the refresher parameter in the Session-Expires: header on the originating call leg that the Oracle Enterprise Session Border Controller includes in the 200 OK response message. Valid values are uac and uas.
9. response-refresher—Set this to the value to insert in the refresher parameter in the Session-Expires: header on the terminating call leg that the Oracle Enterprise Session Border Controller includes in the INVITE message. Valid values are uac, uas, none.
10. Type done to save your work and continue.

    To apply a session timer profile to a SIP interface:

11. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

12. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

13. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

    If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

14. session-timer-profile—Enter the name of a session timer profile object you have configured and want applied to this SIP interface.
15. Type done to save your work and continue.

Verify Config Validation

The Oracle Enterprise Session Border Controller's verify-config function checks that the value configured in all sip-interfaces' session-timer-profiles correspond to a configured session-timer-profile name. The following is be generated when this check fails:

```
ERROR: sip-interface [id] has reference to session-timer-profile [xyz] which
does not exist
```

show sipd status

The show sipd status command now contains new statistic, called Refreshes Sent which reflects the number of refresh requests that the Oracle Enterprise Session Border Controller has sent. For example:

```
ACMEPACKET#show sipd status
SIP Status                -- Period -- -------- Lifetime --------
                  Active    High   Total       Total  PerMax     High
Sessions               0       1       0           2       1        1
Subscriptions          0       0       0           0       0        0
Dialogs                0       2       0           4       2        2
CallID Map             0       2       0           4       2        2
Rejections             -       -       0           0       0
ReINVITEs              -       -       0           2       1
ReINV Suppress         -       -       0           1       1
Media Sessions         0       1       0           2       1        1
Media Pending          0       0       0           0       0        0
Client Trans           0       3       2          10       3        3
Server Trans           0       0       0           6       3        3
Resp Contexts          0       0       0           6       3        3
Saved Contexts         0       0       0           0       0        0
Sockets                2       2       0           2       2        2
Req Dropped            -       -       0           0       0
Refreshes Sent         0       0       0           0       0        0
DNS Trans              0       0       0           0       0        0
DNS Sockets            0       0       0           0       0        0
DNS Results            0       0       0           0       0        0
Rejected Msgs          0       0       0           0       0        0
```

9

Session Recording

SelectiveCall Recording SIPREC

The SIPREC protocol is the protocol used to interact between a Session Recording Client (SRC) (the role performed by Oracle Enterprise Session Border Controller) and a Session Recording Server (SRS) (a 3rd party call recorder or Oracle Communications Interactive Session Recorder's Record and Store Server (RSS)). It controls the recording of media transmitted in the context of a communications session (CS) between multiple user agents.

SIPREC provides a selective-based call recording solution that increases media and signaling performance on 3rd party call recording servers, more robust failovers, and the ability to selectively record.

Note: SIPREC isolates the 3rd party recorders from the communication session. The 3rd party recorders can determine whether or not recording is desired.

Note: The SRC starts a recording session for every call within a configured realm. All call filtering, if desired, must be accomplished by the SRS. The SRS performs the filtering and selection of which sessions it should record.

SIPREC Feature

The SIPREC feature supports active recording, where the Oracle Enterprise Session Border Controller acting as the SRC, purposefully streams media to the Oracle Communications Interactive Session Recorder's RSS (or 3rd party call recorder) acting as the SRS. The SRC and SRS act as SIP User Agents (UAs). The SRC provides additional information to the SRS to describe the communication sessions, participants and media streams for the recording session to facilitate archival and retrieval of the recorded information.

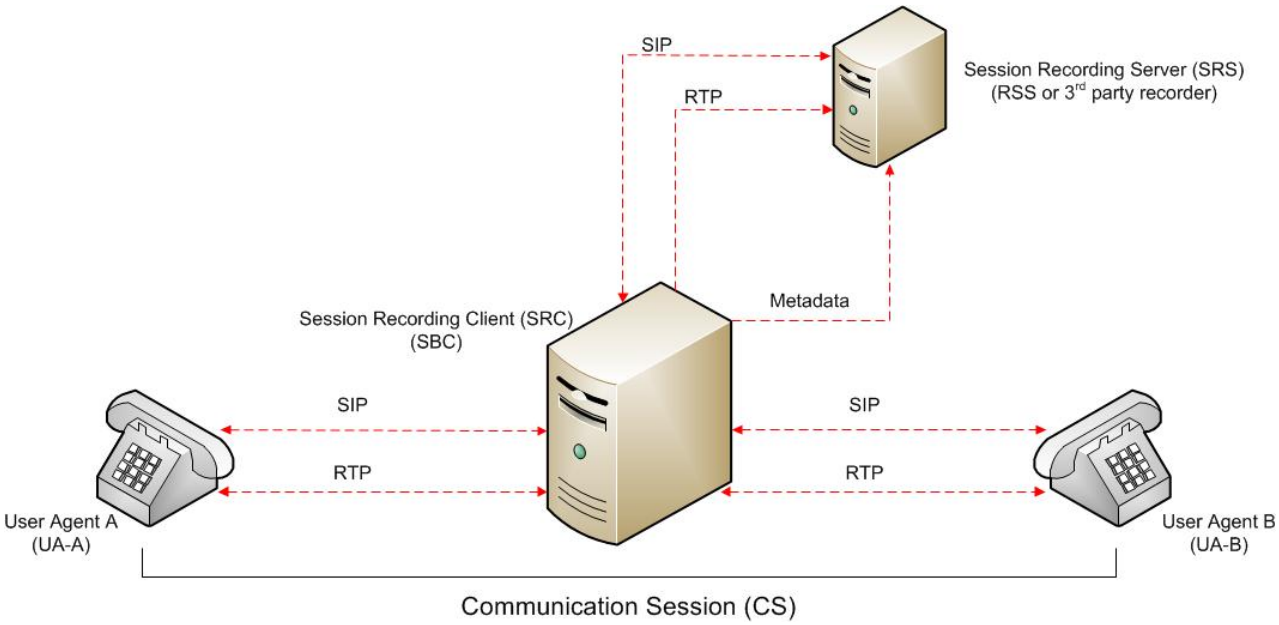
The Oracle Enterprise Session Border Controller acting as the SRC, is the source for the recorded media. The Oracle Enterprise Session Border Controller consumes configuration information describing the ecosystem within which it operates. The interface, realm and session agent configuration objects specify the SIPREC configuration. A SIP UA can elect to allow or disallow any network element from recording its media.

During the establishment of a SIP Session, the Oracle Enterprise Session Border Controller determines if SIPREC is configured for recording the call. If so, it then duplicates the media prior to initiating the session with the SRS. (Media replication is set up prior to the recording session). The SRS may choose to record, not record, or cancel the recording session, and then communicates via SIP signaling to the Oracle Enterprise Session Border Controller. If the call is not to be recorded, the SRS signals termination of the recording session.

The Oracle Enterprise Session Border Controller maintains SIPREC metadata information associated with recording sessions. The recording session metadata describes the current state of the recording session and its communication session(s). It is updated when a change of state in the communication session(s) is observed by the Oracle Enterprise

Session Border Controller. The SRS is responsible for maintaining call history, etc. The Oracle Enterprise Session Border Controller creates and logs call detail records (CDRs) in the current manner, the 3rd party SRS vendor may collate this information if desired. (For more information about the contents of metadata, see *Metadata Contents*).

The following illustration shows two endpoints, User Agent A (UA-A) and User Agent B (UA-B). Their session is being recorded by an SRC (the Oracle Enterprise Session Border Controller) and an SRS.



Configuring SIPREC

This section defines the information required to configure SIPREC on the Oracle Enterprise Session Border Controller. It also provides a sample procedure for configuring SIPREC using the Acme Packet Command Line Interface (ACLI).

Session Recording Server (SRS)

The Oracle Communications Interactive Session Recorder's RSS acts as the SRS in the network. A session-recording-server attribute under the session-router object in the Oracle Enterprise Session Border Controller ACLI allows you to enable/disable the SRS. This object is the session recording server that receives replicated media and records signaling. Additional parameters for SRS are configured under the session-agent, realm-config, and sip-interface objects. The rules of precedence for which the Oracle Enterprise Session Border Controller uses these parameters are: session-agent takes precedence over the realm-config, and realm-config takes precedence over sip-interface.

Each SRS is associated with a realm-config. The realm specifies the source interface from which replicated traffic originates. The destination is an IP Port parameter (IP address or hostname with an optional port) that defines the SIP address (request URI) of the actual SRS.

For an additional level of security, Oracle recommends the SRS be configured in its own realm so as to apply a set of access control lists (ACLs) and security for the replicated communication.

Although the Oracle Enterprise Session Border Controller supports large UDP packets, Oracle recommends the sip-interface associated with the SRS realm, be provisioned with a TCP port.

Session Recording Group

The Oracle Enterprise Session Border Controller uses the session-recording-group attribute under the session-router object in the ACLI to set high availability (HA) for 3rd party call recorders. Using this object, you can define a collection of one or more SRSs. The Oracle Enterprise Session Border Controller utilizes SIP's transport mechanism and keeps track of statistics on each SRS to manage the distribution of traffic and load balancing. (For more information on Oracle Enterprise Session Border Controller load balancing in session recording groups, see *Load*

Balancing). When multiple SRSs are in a session recording group, the Oracle Enterprise Session Border Controller uses heuristics to intelligently route the recording dialog to one or more SRSs utilizing the selection strategy.

The simultaneous-recording-servers configuration attribute controls the number of simultaneous SIP dialogs that the Oracle Enterprise Session Border Controller establishes to the SRSs in the session recording group per communication session. For instance, if a session recording group contains 3 SRSs, and simultaneous-recording-servers is set to 2, the recording agent initiates a SIP INVITE to the next two SRSs based on the session recording group strategy. In this way, duplicative recording sessions are instantiated, allowing for recording redundancy in multiple SRSs or within a session recording group.

Note: The Oracle Enterprise Session Border Controller streams media to all SRSs. Each SRS chooses whether or not to ignore the media by returning a recvonly(receive only) media line. This permits an SRS to select specific media to record in the recording session, as well as determine whether or not to record the media.

The number of simultaneous recording servers does not dictate the number of recording devices required to be active for a communication session. If two SRSs exist in a session recording group and simultaneous-recording-servers is set to 2, if at least one recording device to any of the servers completes, the recording server is treated as being established.

Load Balancing

The Oracle Enterprise Session Border Controller supports recording server load balancing across members of a session recording group using the following strategies:

Note: SRS groups support “round-robin” and “hunt” strategies only.

[Round-robin]: The Oracle Enterprise Session Border Controller remembers the last SRS that was used. Each new recording session selects the next SRS in the session recording group. When simultaneous-recording-servers is greater than 1, the next n recording servers are selected from the session recording group.

[hunt]: The Oracle Enterprise Session Border Controller successively attempts to contact SRSs in the session recording group until a successful recording dialog is established with the SRS, starting from the first SRS in the session recording group. The Oracle Enterprise Session Border Controller attempts to contact each SRS in the session reporting group once. When contact is exhausted, the recording device is considered failed. A SIP failure (response greater than 399, timeout or TCP setup failure) causes the Oracle Enterprise Session Border Controller to attempt the next possible SRS. When simultaneous-recording-servers is greater than 1, the Oracle Enterprise Session Border Controller attempts to establish n recording devices in a hunting fashion.

Session Recording Group within Logical Remote Entities

Each logical remote entity (session-agent, realm-config and sip-interface) has a session-recording-server attribute.This attribute is a reference to a specific SRS configuration and can be used to specify a session recording group instead. If a session recording group is specified instead of an SRS, the session recording group name must be prefixed with "SRG:" followed by the session recording group name. This distinguishes between an SRS being referenced and a session recording group being referenced.

With SIPREC, if an SRS or session recording group is configured on both the ingress and egress logical remote entities, both the ingress and egress SRS/session recording groups are used. This means that the Oracle Enterprise Session Border Controller records the media between participants twice (or more) - once for the ingress recorders and once for the egress recorders.

If both the ingress and egress SRS/session recording group are the same, the Oracle Enterprise Session Border Controller makes an optimization and only records the media once. Even if the ingress session recording group is the same exact set of SRSs as the egress session recording group (but with a different name), the Oracle Enterprise Session Border Controller replicates media to both destinations. However, if the same set of SRSs has the exact same identifier, the
Oracle Enterprise Session Border Controller sends media to one and not both SRSs.

Selective Recording

SIPREC defines a number of use cases for which the Oracle Enterprise Session Border Controller can record communication sessions. These use cases include the use of selective based recording. A selective recording is one in which a unique recording server is created per communication session.

Note: The Oracle Enterprise Session Border Controller does not support persistent recording.

For SRSs using selective recording, recording servers are unique per session recording group. For each selective SRS in a session recording group, during the setup of a new communication session, the recording metadata is the same for each recording device. The SRC initiates a new SIP INVITE to the SRS carrying the metadata for that new recording server. The recording agent terminates the SIP dialog at the time that the recording session ends.

The lifetime of a recording session extends beyond the lifetime of the recorded communication. The SRC (Oracle Enterprise Session Border Controller) re-uses the recording session ID in the metadata instead of creating a new ID for each recording.

High Availability (HA) Support

An Oracle Enterprise Session Border Controller using SIPREC supports HA in the network. The Oracle Enterprise Session Border Controller replicates all metadata states between the active and standby Oracle Enterprise Session Border Controllers. Any recording dialogs in progress do not survive the failover, but all calls in progress are preserved. Additionally, the recording dialogs are replicated as well to the failed over Oracle Enterprise Session Border Controller so that in-dialog SIP requests continue to function.

Each recorded communication session replicated to a single SRS counts as two calls instead of one. The Oracle Enterprise Session Border Controller creates two flows between the two participants and two additional flows to the SRS for each of the parent flows.

Single SRS

Assuming that each communication session (CS) is recorded to a single SRS with a single recording session, the total session capacity for recorded sessions is as follows:

- NN4500: 4,000 sessions
- Server Edition: 250 sessions
- VM Edition: 125 sessions

SIPREC Configuration Procedure

The following configuration example assumes the Oracle Enterprise Session Border Controller has the session recording license enabled on the Oracle Enterprise Session Border Controller. Changes to the call session recording configuration for SIPREC are dynamic. Active calls in progress remain unaffected by the configuration changes. New calls, however, utilize the changes after a Save and Activate of the configuration.

The following attributes must be configured:

- session-recording-server
- session-recording-group (for RSS or 3rd party SRS high availability (HA) only)

and at least one of the following attributes:

- realm-config
- session-agent
- sip-interface

Session-recording-server Attribute

To configure the session-recording-server attribute:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-recording-server and press Enter to access the session recording server-related attributes.

```
ACMEPACKET(session-router)# session-recording-server
ACMEPACKET(session-recording-server)#
```

4. name — Enter a unique name for the session recording server. This name can be referenced when configuring realm-config, session-agent, and sip-interface. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# name SRS1
```

5. (optional) description — Enter a description for the session recording server. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# description <recording server name>
```

6. realm — Enter the realm for which the session recording server belongs. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# realm <realm name>
```

Note: Oracle recommends that the session recording server be configured in its own realm.

7. mode — Enter the recording mode for the session recording server. Valid values are:

   - selective (default) - Unique recording server created per communication session
   - persistent - Not supported.

```
ACMEPACKET(session-recording-server)# recording-mode selective
```

8. destination — Enter the destination IP address with IP port (port specification is optional) that defines the SIP address (request URI) of the session recording server. Enter values in the format 0.0.0.0:<port number>. Default is no value specified.

```
ACMEPACKET(session-recording-server)# destination 172.34.2.3:5060
```

9. port — Enter the port number to contact the session recording server. Valid values are 1024 to 65535. Default is 5060.
10. transport-method — Enter the protocol that the session recording server uses to accept incoming packets from the session reporting client on the network. Default is DynamicTCP. Valid values are:

    - “” - No transport method used. Same as leaving this parameter value blank.
    - UDP - User Datagram Protocol (UDP) is used for transport method.
    - UDP+TCP - UDP and Transmission Control Protocol (TCP) are used for transport method.
    - DynamicTCP - One TCP connection for EACH session is used for the transport method.
    - StaticTCP - Only one TCP connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.
    - DynamicTLS - One Transport Layer Security (TLS) connection for EACH session is used for the transport method.
    - StaticTLS - Only one TLS connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.
    - DTLS - Datagram TLS is used for the transport method.
    - TLS+DTLS - TLS and DTLS are used for the transport method.
    - StaticSCTP - Only one Stream Control Transmission Protocol (SCTP) connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.

```
ACMEPACKET(session-recording-server)# protocol UDP
```

11. Enter done to save the session recording configuration.

```
ACMEPACKET(session-recording-server)# done
```

12. Enter exit to exit the session-recording-server configuration.

```
ACMEPACKET(session-recording-server)# exit
```

13. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

14. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

15. Enter save-config to save the session recording configuration.

```
ACMEPACKET# save-config
```

16. Enter activate-config to activate the session recording configuration.

```
ACMEPACKET# activate-config
```

Session-recording-group Attribute (for HA only)

For environments that required high availability (HA) requirements, configure the session-recording-group attribute.

To configure the session-recording-group attribute and enable HA:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-recording-group and press Enter to access the session recording group-related attributes.

```
ACMEPACKET(session-router)# session-recording-group
ACMEPACKET(session-recording-group)#
```

4. name — Enter a unique name for the session recording group that is a collection of one or more session recording servers. This name can be referenced when configuring realm-config, session-agent, and sip-interface. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-group)# name <SRG Group Name>
```

Note: The name of the session recording group must be prefixed with SRG.

5. (optional) description — Enter a description for the session recording group. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-group)# description <Recording Group Name>
```

6. session-recording-servers — Enter the names of the session recording servers that belong to this session recording group. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-group)# session-recording-servers SRS1,SRS2
```

Note: You must enter multiple servers as values for the session-recording-servers attribute.

7. strategy — Enter the load balancing strategy that the session reporting client (Oracle Enterprise Session Border Controller) uses when sending recordings to the session reporting server. Valid values are:

   - Round-robin (default) - The Oracle Enterprise Session Border Controller remembers the last SRS that was used. Each new recording session selects the next SRS in the session recording group. When simultaneous-recording-servers is greater than 1, the next n recording servers are selected from the session recording group.
   - hunt - The Oracle Enterprise Session Border Controller successively attempts to contact SRSs in the session recording group until a successful recording dialog is established with the SRS, starting from the first SRS in the session recording group. The Oracle Enterprise Session Border Controller attempts to contact each SRS in the session reporting group once. When contact is exhausted, the recording device is considered failed. A SIP failure (response greater than 399, timeout or TCP setup failure) causes the Oracle Enterprise Session Border Controller to attempt the next possible SRS. When simultaneous-recording-servers is greater than 1, the Oracle Enterprise Session Border Controller attempts to establish n recording devices in a hunting fashion.

- least busy - For some 3rd party recording devices, the number of concurrent recording servers proves to be the most taxing for system resources. The Oracle Enterprise Session Border Controller tracks the number of recording servers active to a given SRS at any given time. It uses this information to determine which SRS would be the best candidate for the next RS. The SRS with the fewest number of active recording servers receives the next RS. If two or more SRSs in a session recording group currently have the same number of active recording servers, the SRS configured first in the session recording group takes precedence.
- lowest sustained rate (fewest-setups-per-minute) - For some 3rd party recording servers, processing large amounts of sessions in a short amount of time proves to be the most taxing on their system's resources. The Oracle Enterprise Session Border Controller tracks the number of recording server setups over a sliding window of five minutes. The SRS within the session recording group with the fewest setups per the window of time is selected as the next candidate for receiving the recorded session. If two or more SRSs in a session recording group currently have the same value for setups in the given window of time, then the SRS configured first in the session recording group takes precedence.

```
ACMEPACKET(session-recording-group)# strategy round-robin
```

8. simultaneous-recording-servers — Enter the number of simultaneous SIP dialogs that the session reporting client (Oracle Enterprise Session Border Controller) establishes to the session reporting servers in the session reporting group per communication session. Valid values are 1 to 3. Default is 0.

```
ACMEPACKET(session-recording-group)# simultaneous-recording-servers 2
```

9. Enter done to save the session recording group configuration.

```
ACMEPACKET(session-recording-group)# done
```

10. Enter exit to exit the session recording group configuration.

```
ACMEPACKET(session-recording-group)# exit
```

11. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

12. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

13. Enter save-config to save the session recording group configuration.

```
ACMEPACKET# save-config
```

14. Enter activate-config to activate the session recording group configuration.

```
ACMEPACKET# activate-config
```

Realm-config Attribute

To configure the realm-config attribute and enable session recording:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media manager-related objects.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter to access the realm-config-related attributes.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. session-recording-server — Enter the name of the session-recording server or the session-recording-group in the realm associated with the session reporting client (Oracle Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(realm-config)# session-recording-server <srs-name>

or

ACMEPACKET(realm-config)# session-recording-server SRG:<group-name>
```

Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*. If specifying a session-recording-group, you must precede the group name with SRG:.

5. session-recording-required — Enter whether or not you want a call to be accepted by the Oracle Enterprise Session Border Controller if recording is not available. Valid values are:

   - Enabled - Restricts call sessions from being initiated when a recording server is not available.
   - Disabled (default)- Allows call sessions to initiate even if the recording server is not available.

   ```
   ACMEPACKET(realm-config)# session-recording-required disabled
   ```

   Note: Oracle recommends that the session-recording-required parameter remain disabled.

6. Enter done to save the realm configuration.

   ```
   ACMEPACKET(realm-config)# done
   ```

7. Enter exit to exit the realm configuration.

   ```
   ACMEPACKET(realm-config)# exit
   ```

8. Enter exit to exit the media manager configuration.

   ```
   ACMEPACKET(media-manager)# exit
   ```

9. Enter exit to exit the configure mode.

   ```
   ACMEPACKET(configure)# exit
   ```

10. Enter save-config to save the realm configuration.

    ```
    ACMEPACKET# save-config
    ```

11. Enter activate-config to activate the realm configuration.

    ```
    ACMEPACKET# activate-config
    ```

Session-agent Attribute

To configure the session-agent attribute and enable session recording:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session router-related objects.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type session-agent and press Enter to access the session agent-related attributes.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. session-recording-server — Enter the name of the session-recording server or the session-recording-group to apply to the session recording client (Oracle Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

   ```
   ACMEPACKET(session-agent)# session-recording-server <srs-name>
   ```

   or

   ```
   ACMEPACKET(session-agent)# session-recording-server SRG:<group-name>
   ```

   Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*. If specifying a session-recording-group, you must precede the group name with SRG:.

5. session-recording-required — Enter whether or not you want a call to be accepted by the Oracle Enterprise Session Border Controller if recording is not available. Valid values are:

   - Enabled - Restricts call sessions from being initiated when a recording server is not available.
   - Disabled (default)- Allows call sessions to initiate even if the recording server is not available.

```
ACMEPACKET(session-agent)# session-recording-required disabled
```

Note: Oracle recommends that the session-recording-required parameter remain disabled.

6. Enter exit to exit the session agent configuration.

```
ACMEPACKET(session-agent)# exit
```

7. Enter exit to exit the session router configuration.

```
ACMEPACKET(session-router)# exit
```

8. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

9. Enter save-config to save the session agent configuration.

```
ACMEPACKET# save-config
```

10. Enter activate-config to activate the session agent configuration.

```
ACMEPACKET# activate-config
```

Sip-interface Attribute

To configure the sip-interface attribute and enable session recording:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter to access the SIP interface-related attributes.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. session-recording-server — Enter the name of the session-recording server or the session-recording-group to apply to the SIP interface on the session recording client (Oracle Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(sip-interface)# se ss on-recording-server SRG:<session recording
server name or session-recording group name>
```

Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*.

5. session-recording-required — Enter whether or not you want a call to be accepted by the Oracle Enterprise Session Border Controller if recording is not available. Valid values are:

- Enabled - Restricts call sessions from being initiated when a recording server is not available.
- Disabled (default)- Allows call sessions to initiate even if the recording server is not available.

```
ACMEPACKET(sip-interface)# session-recording-required disabled
```

Note: Oracle recommends that the session-recording-required parameter remain disabled.

6. Enter exit to exit the SIP interface configuration.

```
ACMEPACKET(sip-interface)# exit
```

7. Enter exit to exit the session router configuration.

```
ACMEPACKET(session-router)# exit
```

8. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

9. Enter save-config to save the SIP interface configuration.

```
ACMEPACKET# save-config
```

10. Enter activate-config to activate the SIP interface configuration.

```
ACMEPACKET# activate-config
```

SIPREC Ping

This SIPREC ping is a signal that the Oracle Enterprise Session Border Controller transmits to the connected SRS requesting a response pertaining to the message type that you specify for the ping-method. It uses the ping-interval to determine how long it should wait before sending another ping to the SRS.

You can check the connectivity by configuring the following parameters:

- Ping method- SIP message or method for which to ping the SRS.
- Ping interval- Amount of time, in seconds, that the Oracle Enterprise Session Border Controller waits before it pings the SRS in subsequent intervals. For example, if this parameter is set for 60 seconds, the Oracle Enterprise Session Border Controller pings the SRS every 60 seconds.

Once configured the Oracle Enterprise Session Border Controller uses this feature to perform SIP-based pinging to determine if the SRS is reachable or not.

Configuring SIPREC Ping on theOracle Enterprise Session Border Controller.

To configure SIPREC ping on the Oracle Enterprise Session Border Controller, you use the ping-method and the ping-interval objects under call-recording-server. Use the following procedure to configure SIPREC ping on the Oracle Enterprise Session Border Controller.

To configure SIPREC ping:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type call-recording-server and press Enter.

```
ACMEPACKET(session-router)# call-recording-server
ACMEPACKET(call-recording-server)#
```

4. ping-method—Enter the message or method type for which the Net-Net ESD uses in a ping request to the SRS to determine if it is reachable or not. Default is blank. Valid values are:

BYE	OPTIONS
UPDATE	SUBSCRIBE
CANCEL	NOTIFY

5. ping-interval—Enter the amount of time, in seconds, that the Net-Net ESD waits before it pings the SRS in subsequent intervals. Valid values are 0 to 99999. Default is 0 (zero). The setting of zero disables the ping interval.
6. Type done and press Enter.

```
ACMEPACKET(call-recording-server)# done
ACMEPACKET(call-recording-server)#
```

7. Type exit and press Enter.

```
ACMEPACKET(call-recording-server)# exit
ACMEPACKET(session-router)#
```

8. Type exit and press Enter.

```
ACMEPACKET(session-router)# exit
ACMEPACKET(configure)#
```

9. Save the configuration.

Example SIPREC Ping Configuration

The following is an example of a SIPREC ping configuration.

```
call-recording-server# show
     name                  SRS1
     description           session recording server
     realm                 realmA
     mode                  selective
     destination           132.43.5.6
     port                  5060
     transport-method      DynamicTCP
     ping-method           OPTIONS
     ping-interval         60
```

In the above example, the Net-Net ESD sends a ping request to the SRS using the OPTIONS value every 60 seconds to determine if the SRS is reachable or not.

Metadata Contents

The recording metadata contains a set of related elements which define the recording session. A recording session may contain zero or more communication sessions and/or communication session groups. A communication session represents a call instance; a communication session group represents a related group of communication sessions. A recording session is composed of a sequence of complex element types. Not all element types are required to describe a recording session initiated from the Oracle Enterprise Session Border Controller. The recording session XML schema defines the following element types:

- dataMode - partial or complete metadata description (required)
- group - a collection of related communication sessions
- session - a single communication session of two or more participants (required)
- participant - a SIP endpoint representation (required)
- stream - a media stream
- extensiondata - application specific data outside of the SIPREC scope.

The recording agent generates dataMode, session, participant, and stream elements. Extension data is attached to other elements within the metadata through the use of the parent attribute. The recording metadata is defined as a sequence of element types; therefore all associations between elements are represented as references to element identifiers.

The state of the metadata within a recording session reflects the state of the communication session(s) which is being recorded. SIPREC implements stop-times and reason codes when communication sessions end within a recording session. Once a communication session, participant, or media stream has been marked as 'stopped' and accepted by the SRS, the metadata item is removed from the current metadata state. In addition, media lines within the SDP or the recording session may be re-used/re-labeled for reuse if new communication sessions and media streams are created within the recording session.

The XML schema for the recording metadata is defined in the IETF draft RFC *draft-ram-siprec-metadata-format-02* [7].

The ACLI command to show recorded metadata is show rec. For more information on this command see the section, *Show rec*.

Show Commands for Recording Sessions

The Oracle Enterprise Session Border Controller allows you to utilize the following show commands via the ACLI to display statistical information about recording sessions:

- show rec
- show rec redundancy

Show rec

The show rec command displays the count of all metadata objects in sessions managed by the recording agent. These statistics include metadata monitored over an active period of time and over a lifetime period (where lifetime totals reflect from the last reboot of the Oracle Enterprise Session Border Controller to the present time). The following example shows the use of this command.

1. Log into the Oracle Enterprise Session Border Controller as a User or Superuser.

```
ACMEPACKET> enable
ACMEPACKET(enable)#
```

2. Type show rec and press Enter to display the recording metadata statistics. The following output is an example of the show rec command.

```
ACMEPACKET(enable)# show rec
```

Show rec output

```
13:49:44-81645
Recording Agent Status        -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2
```

The following table describes the metadata objects in the show rec command output.

Object	Description
Rec Sessions	Number of recording sessions during an active period of time and over a lifetime period.
Comm Groups	Number of active communication session recording groups during an active period of time and over a lifetime period.
Comm Sessions	Number of active communication sessions during an active period of time and over a lifetime period.
Media Streams	Number of active media streams during an active period of time and over a lifetime period.
Participants	Total number of participants in session recordings during an active period of time and over a lifetime period.

Show rec redundancy

The show rec redundancy command displays information for session recording server statistics when the Oracle Enterprise Session Border Controller is configured for HA. These statistics include metadata monitored over an active period of time and over a lifetime period (where lifetime totals reflect from the last reboot of the Oracle Enterprise Session Border Controller to the present time) on both the primary and redundant Oracle Enterprise Session Border Controller. The following example shows the use of this command.

1. Log into the Oracle Enterprise Session Border Controller as a User or Superuser.

```
ACMEPACKET> enable
ACMEPACKET(enable)#
```

2. Type show rec redundancy and press Enter to display the session recording server statistics for Oracle Enterprise Session Border Controllers in HA mode. The following output is an example of the show rec redundancy command.

```
ACMEPACKET(enable)# show rec redundancy
```

Show rec redundancy output

Primary System

```
13:49:44-81645
Recording Agent Status         -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2

Redundant System
13:49:44-81646
Recording Agent Status         -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2
```

The following table describes the session recording server statistics in the show rec redundancy command output.

Object	Description
Rec Sessions	Number of recording sessions during an active period of time and over a lifetime period.
Comm Groups	Number of active communication session recording groups during an active period of time and over a lifetime period.
Comm Sessions	Number of active communication sessions during an active period of time and over a lifetime period.
Media Streams	Number of active media streams during an active period of time and over a lifetime period.
Participants	Total number of participants in session recordings during an active period of time and over a lifetime period.

Codec Negotiation

In a SIPREC environment, it is assumed that the recording ecosystem provides transcoding media servers for which media calls can be redirected to, relieving the issue of codec matching from the recording servers. However, if transcoding media servers are not provided, the responsibility for transcoding falls on the recording server or the recording client in a SIPREC environment. The Oracle Enterprise Session Border Controller/SRC is required to impose some policy decisions on the codec negotiation between the three, or more, end-points. Specifically, the codec negotiation between the two participants and the recording server is subject to additional policy actions.

The SDP answer from the SRS may not agree with the media flows established in the communication session between UA-A and UA-B. If UA-A and UA-B agree to use G729, yet the SRS's answer indicates no support for G729, the SRS is then unable to interpret the media streams. The SDP offer forwarded to the called party (in this case UA-B) limits the codec choices to those supported by the SRS.

Note: The recording agent forwards the original codec offer to the SRS prior to sending the invite to the UA-B. The SRS responds with the SDP answer, indicating the codec list most desirable to the SRS. The codec list in the answer is then forwarded to UA-B. This allows three parties in a conference call to participate in the negotiation of the codecs among the supported formats only.

SIPREC Call Flows

This section provides examples of call flow scenarios that can occur in a SIPREC environment. SIP recording call flow examples include:

For Selective Recording:

- *Normal Call (recording required)*
- *Normal Call (recording not required)*
- *Early Media Call (recording not required)*
- *REFER Pass-Through Call (REFER handled by User Agent)*
- *REFER Call (REFER handled by the Oracle Enterprise Session Border Controller)*
- *SRS Indicates Busy in Call (recording not required)*

Note: REFER is a SIP method indicating that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request.

SIPREC Re-INVITE Collision and Back-off Support

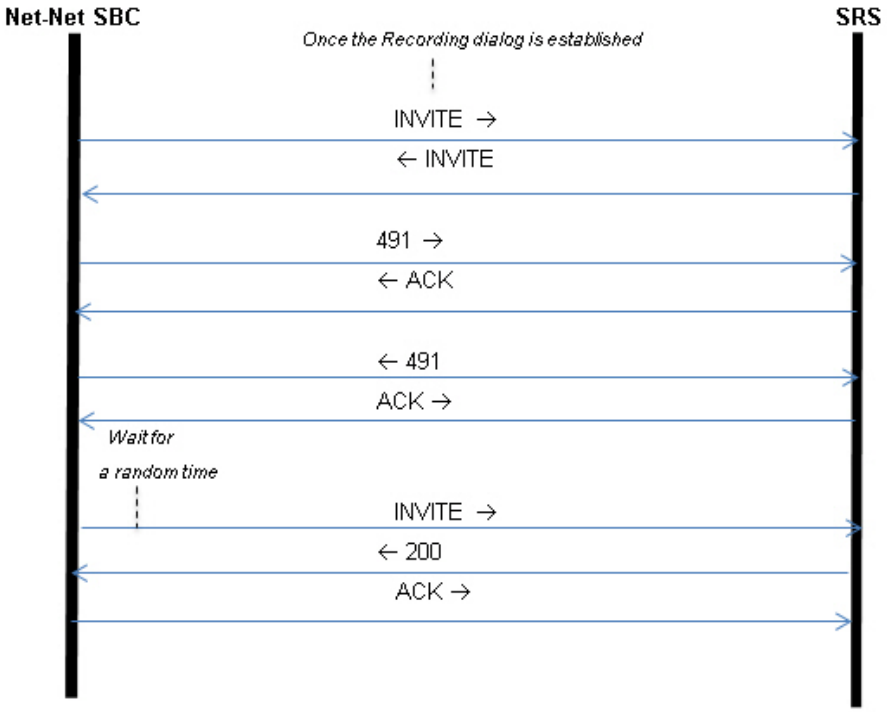
The Oracle SBC acts a back-to-back User Agent (B2BUA) in all call scenarios. However with SIPREC, the Oracle SBC acts as a User Agent Client (UAC) when connected with a session recording server (SRS). Therefore, SIP requests can originate from the Oracle SBC.

During a recording session, when the SRS establishes a recording dialog, the Oracle SBC and the SRS may send Re-INVITES to each other with updated information. When the Oracle SBC receives an INVITE while it is still waiting for the response to a previous INVITE it sent out, this produces an INVITE collision.

To avoid an INVITE collision, the Oracle SBC now sends a 491 Request Pending response back to the SRS and then waits for a random amount of time before re-trying the INVITE. It also acknowledges (ACK) any 491 response received from the other side. RFC 3261 and RFC 6141 describes the way the User Agent (UA) resolves the INVITE collision. The random wait time is chosen based on the following guidelines from RFC 3261:

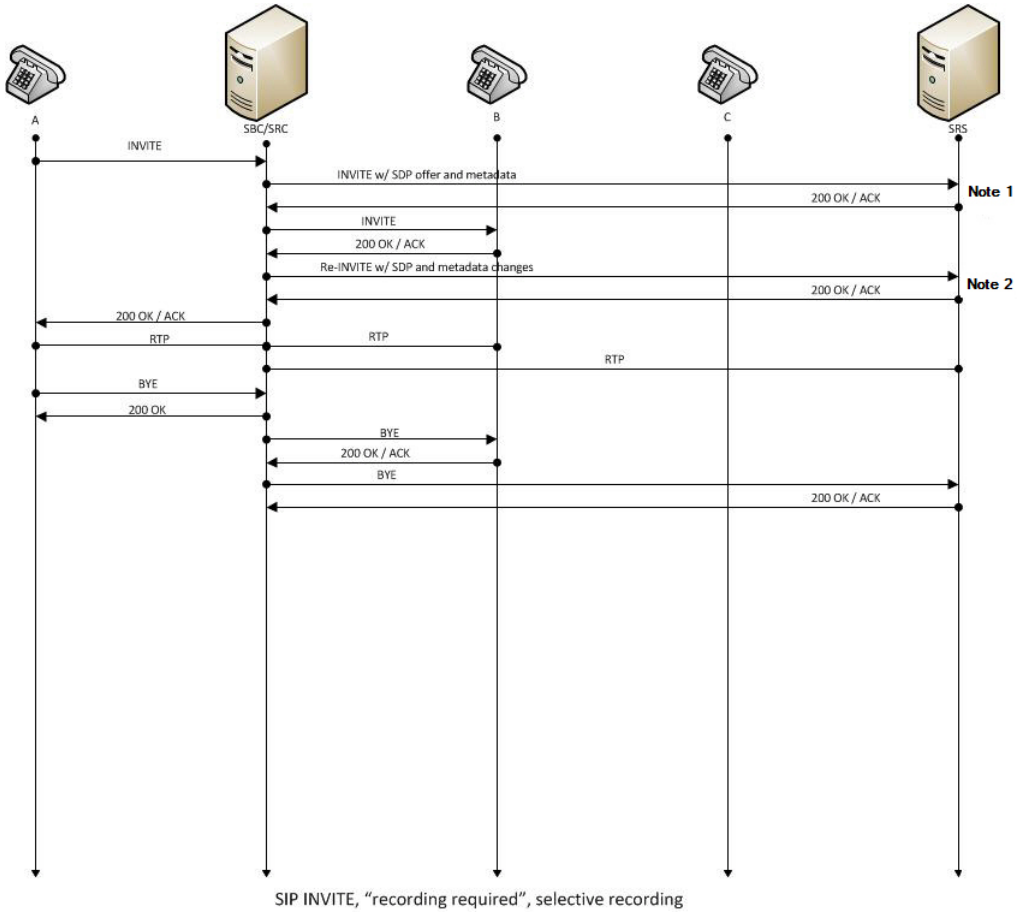
- If the UAC is the owner of the Call-ID of the dialog ID (i.e., it generated the value), T (the wait time) is a randomly chosen value between 2.1 and 4 seconds in units of 10 milliseconds.
- If the UAC is not the owner of the Call-ID of the dialog ID (i.e., it did not generate the value), T (the wait time) is a randomly chosen value between 0 and 2 seconds in units of 10 milliseconds.

The following call flow diagram shows the Oracle SBC's feature to avoid INVITE collision.



Selective Recording
Normal Call (recording required)

The following illustration shows a normal call using selective recording with recording required. For SDP and Metadata information in Notes 1 and 2 , see *Sample SDP and Metadata*.



I

Call Flow Description	
① UA-A sends INVITE to Oracle Enterprise Session Border Controller.	⑩ RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.
② Oracle Enterprise Session Border Controller forwards INVITE with SDP and metadata to SRS.	⑪ RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.
③ SRS responds with OK to Oracle Enterprise Session Border Controller.	⑫ UA-A sends BYE to Oracle Enterprise Session Border Controller.
④ Oracle Enterprise Session Border Controller sends INVITE to UA-B.	⑬ Oracle Enterprise Session Border Controller responds with OK to UA-A.
⑤ UA-B responds with OK to Oracle Enterprise Session Border Controller.	⑭ Oracle Enterprise Session Border Controller sends BYE to Oracle Enterprise Session Border Controller.
⑥ Oracle Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	⑮ Oracle Enterprise Session Border Controller responds with OK to UA-A.
⑦ SRS responds with OK to Oracle Enterprise Session Border Controller.	⑯ Oracle Enterprise Session Border Controller sends BYE to UA-B.

Call Flow Description	
⑧ Oracle Enterprise Session Border Controller forwards OK response to UA-A.	⑰ UA-B responds with OK to Oracle Enterprise Session Border Controller.
⑨ RTP stream initiated between UA-A and Oracle Enterprise Session Border Controller.	⑱ Oracle Enterprise Session Border Controller sends BYE to SRS.
	⑲ SRS responds with OK to Oracle Enterprise Session Border Controller.

Sample SDP and Metadata

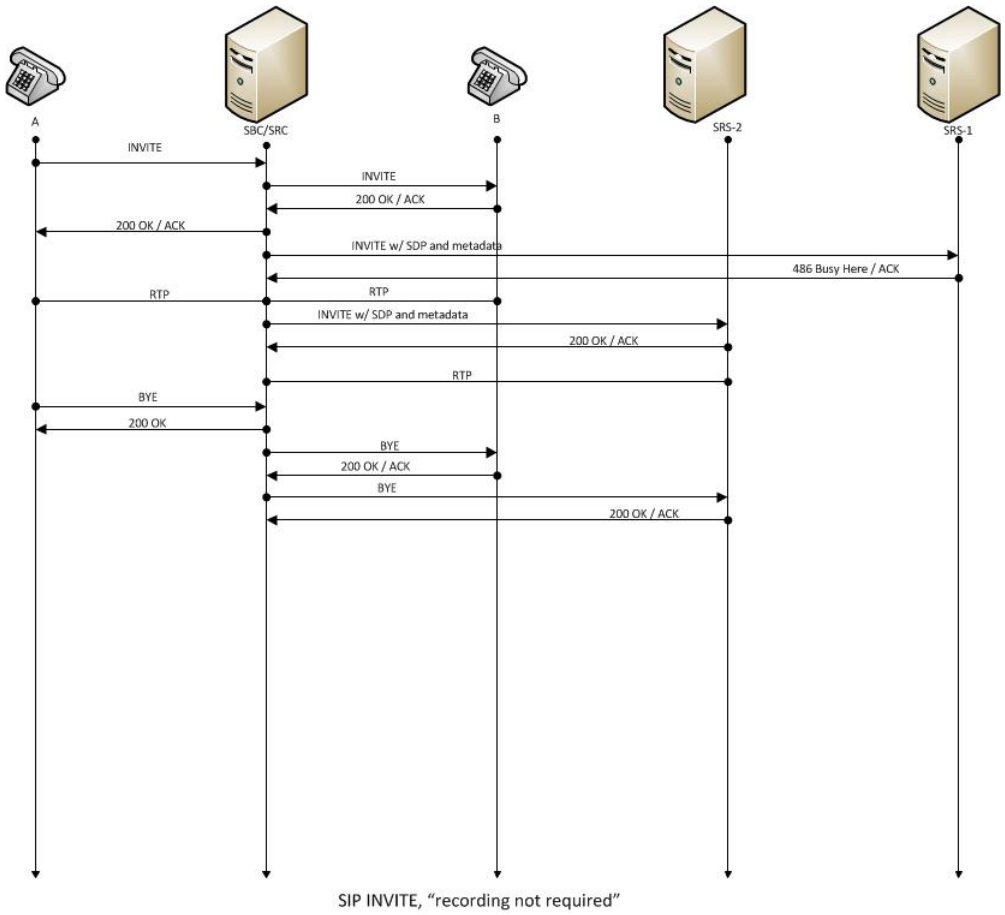
The following sample SDP and Metadata pertain to Notes 1 and 2 in the previous Call Flow diagram.

```
--[Note 1]------------------------------
Content-Type: application/sdp
v=0
o=- 171 213 IN IP4 10.0.0.2
s=-
c=IN IP4 10.0.0.1
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:1

Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
<?xml version='1.0' encoding='UTF-8'?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
        <dataMode>complete</dataMode>
        <session id="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <start-time>2011-06-27T17:03:57</start-time>
        </session>
        <participant id="urn:uuid:10ac9063-76b7-40bb-4587-08ba290d7327"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:sipp@168.192.24.40</aor>
                <name>sipp </name>
                <send>urn:uuid:07868c77-ef8e-4d6f-6dd5-a02ff53a1329</send>
                <start-time>2011-06-27T17:03:57</start-time>
        </participant>
        <participant id="urn:uuid:797c45f5-e765-4b12-52b0-d9be31138529"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:service@168.192.24.60</aor>
                <name>sut </name>
        </participant>
        <stream id="urn:uuid:4a72a1ed-abb2-4d7c-5f4d-6d4c36e2d4ec"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <mode>separate</mode>
                <start-time>2011-06-27T17:03:57</start-time>
                <label>1</label>
        </stream>
</recording>


--[Note 2]------------------------------
Content-Type: application/sdp
v=0
o=- 171 213 IN IP4 10.0.0.2
s=-
c=IN IP4 10.0.0.1
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

```
a=label:1
m=audio 6002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:2

Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
<?xml version='1.0' encoding='UTF-8'?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
        <dataMode>partial</dataMode>
        <session id="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <start-time>2011-06-27T17:03:57</start-time>
        </session>
        <participant id="urn:uuid:797c45f5-e765-4b12-52b0-d9be31138529"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:service@168.192.24.60</aor>
                <name>sut </name>
                <send>urn:uuid:4a72a1ed-abb2-4d7c-5f4d-6d4c36e2d4ec</send>
                <start-time>2011-06-27T17:03:58</start-time>
        </participant>
        <stream id="urn:uuid:07868c77-ef8e-4d6f-6dd5-a02ff53a1329"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <mode>separate</mode>
                <start-time>2011-06-27T17:03:58</start-time>
                <label>2</label>
        </stream>
</recording>
```

Normal Call (recording not required)

The following illustration shows a normal call using selective recording with recording optional.

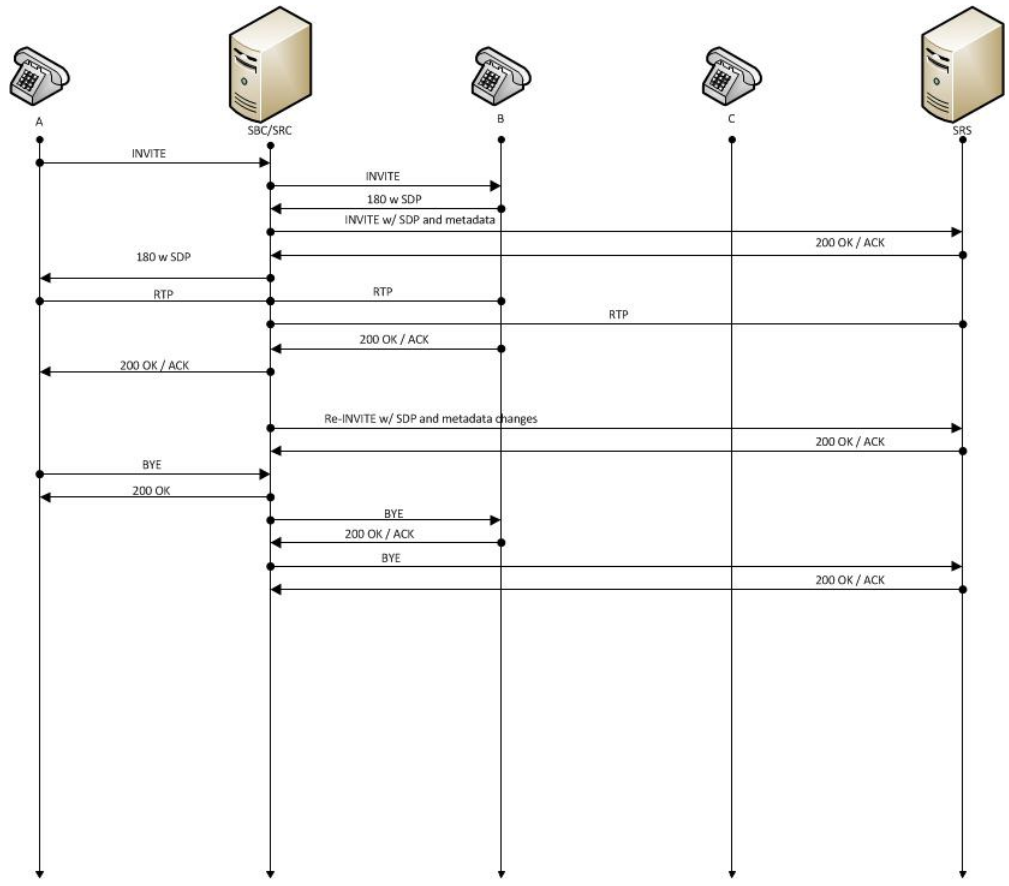


SIP INVITE, "recording not required"

Call Flow Description	
① UA-A sends INVITE to Oracle Enterprise Session Border Controller.	⑧ RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.
② Oracle Enterprise Session Border Controller forwards INVITE to UA-B.	⑨ UA-A sends BYE to Oracle Enterprise Session Border Controller.
③ UA-B responds with OK to Oracle Enterprise Session Border Controller.	⑩ Oracle Enterprise Session Border Controller responds with OK to UA-A.
④ Oracle Enterprise Session Border Controller forwards OK response to UA-A.	⑪ Oracle Enterprise Session Border Controller sends BYE to UA-B.
⑤ Oracle Enterprise Session Border Controller sends INVITE with SDP and metadata to SRS.	⑫ UA-B responds with OK to Oracle Enterprise Session Border Controller.
⑥ SRS responds with OK to Oracle Enterprise Session Border Controller.	⑬ Oracle Enterprise Session Border Controller sends BYE to SRS.
⑦ RTP stream initiated between UA-A, Oracle Enterprise Session Border Controller, and UA-B.	⑭ SRS responds with OK to Oracle Enterprise Session Border Controller.

Early Media Call (recording not required)

The following illustration shows an early media call using selective recording with recording optional.

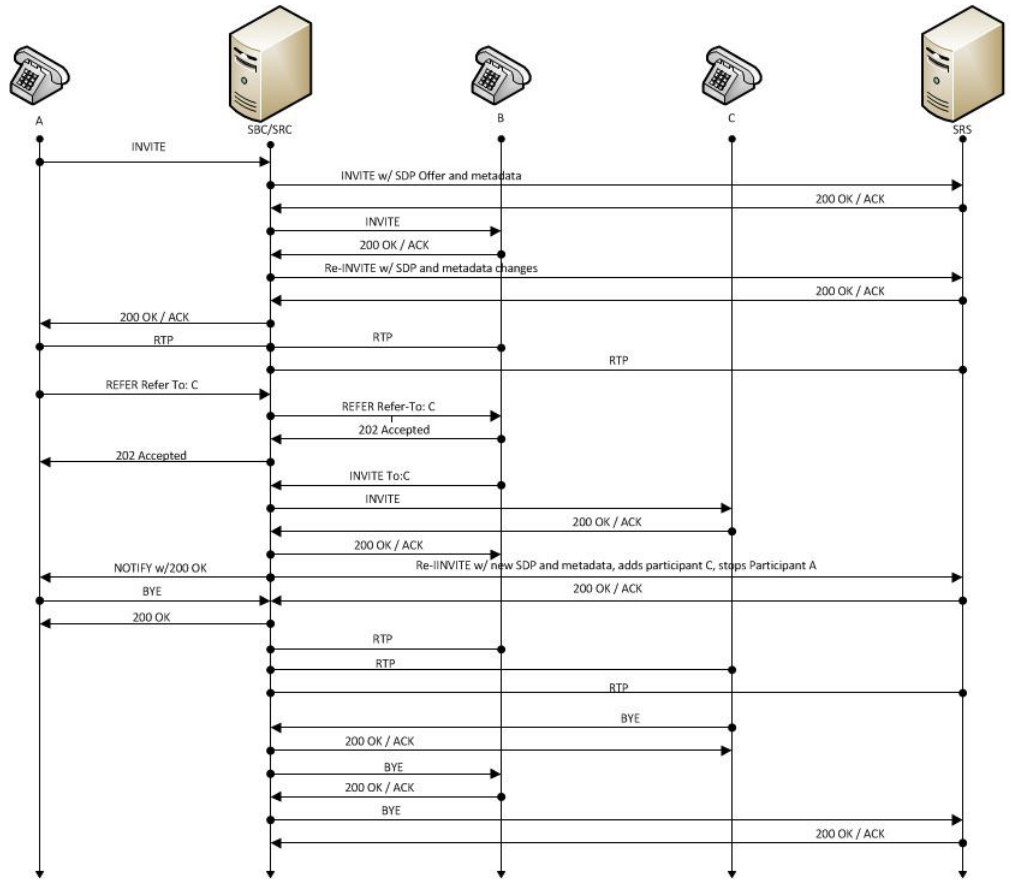


SIP INVITE, early media , "recording not required", selective recording

Call Flow Description	
① UA-A sends INVITE to Oracle Enterprise Session Border Controller.	⑩ UA-B responds with OK to Oracle Enterprise Session Border Controller.
② Oracle Enterprise Session Border Controller forwards INVITE to UA-B.	⑪ Oracle Enterprise Session Border Controller forwards OK to UA-A.
③ UA-B sends 180 and SDP to Oracle Enterprise Session Border Controller.	⑫ Oracle Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.
④ Oracle Enterprise Session Border Controller sends INVITE with SDP and metadata to SRS.	⑬ SRS responds with OK to Oracle Enterprise Session Border Controller.
⑤ SRS responds with OK to Oracle Enterprise Session Border Controller.	⑭ UA-A sends BYE to Oracle Enterprise Session Border Controller.
⑥ Oracle Enterprise Session Border Controller sends 180 with SDP to UA-A.	⑮ Oracle Enterprise Session Border Controller responds with OK to UA-A.
⑦ RTP stream initiated between Oracle Enterprise Session Border Controller and UA-A.	⑯ Oracle Enterprise Session Border Controller sends BYE to UA-B.
⑧ RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.	⑰ UA-B responds with OK to Oracle Enterprise Session Border Controller.
⑨ RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.	⑱ Oracle Enterprise Session Border Controller sends BYE to SRS.
	⑲ SRS responds with OK to Oracle Enterprise Session Border Controller.

REFER Pass-Through Call (REFER handled by User Agent)

The following illustration shows a REFER pass-through call using selective recording and the User Agent (UA) handling the REFER on the call. Recording is required in this call flow.



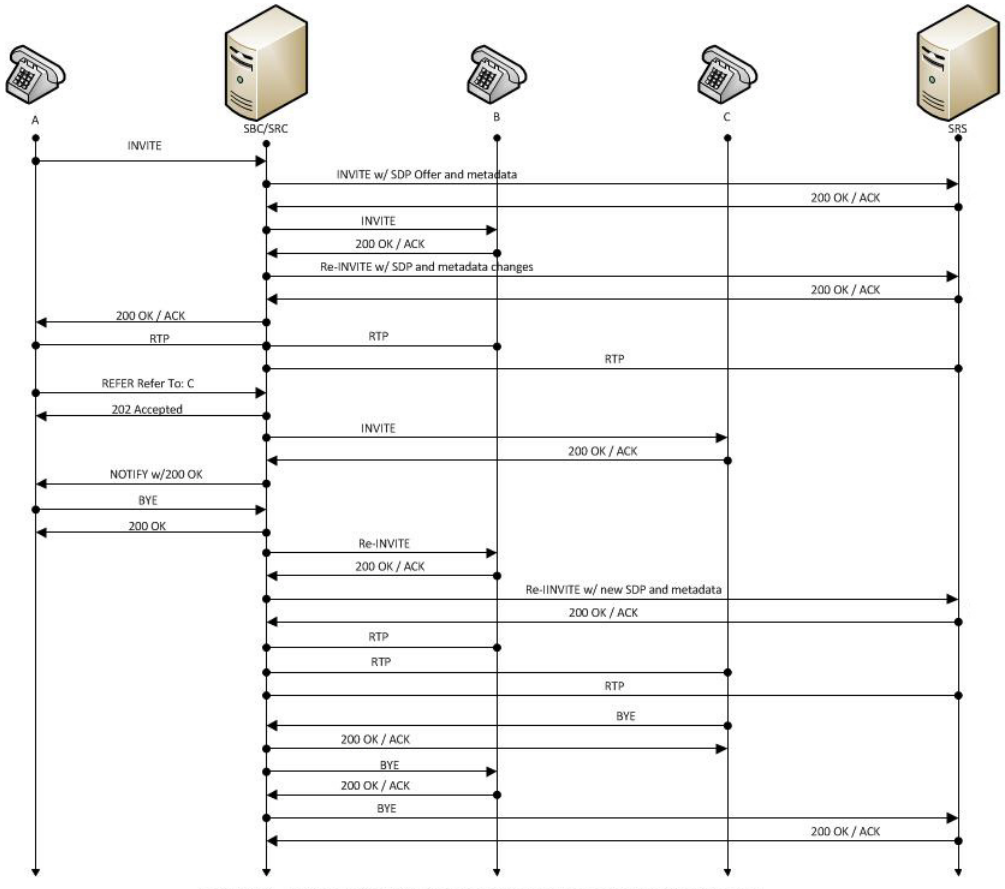
SIP REFER, UA handles REFER, "recording required", selective recording

Call Flow Description	
1 - UA-A sends INVITE to Oracle Enterprise Session Border Controller.	18 - UA-C responds with OK to Oracle Enterprise Session Border Controller.
2 - Oracle Enterprise Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.	19 - Oracle Enterprise Session Border Controller forwards OK response to UA-B.
3 - SRS responds with OK to Oracle Enterprise Session Border Controller.	20 - Oracle Enterprise Session Border Controller sends NOTIFY with OK reponse to UA-A.
4 - Oracle Enterprise Session Border Controller sends INVITE to UA-B.	21 - Oracle Enterprise Session Border Controller sends re-INVITE to SRS with new SDP and metadata, adds participant C, stops participant A .
5 - UA-B responds with OK to Oracle Enterprise Session Border Controller.	22 - SRS responds with OK to Oracle Enterprise Session Border Controller.
6 - Oracle Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	23 - UA-A sends BYE to Oracle Enterprise Session Border Controller.
7 - SRS responds with OK to Oracle Enterprise Session Border Controller.	24 - Oracle Enterprise Session Border Controller responds with OK to UA-A.
8 - Oracle Enterprise Session Border Controller forwards OK response to UA-A.	25 - Oracle Enterprise Session Border Controller responds with OK to UA-A.
9 - RTP stream initiated between UA-A and Oracle Enterprise Session Border Controller.	26 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.

Call Flow Description	
10 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.	27 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-C.
11 - RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.	28 - RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.
12 - UA-A sends REFER-TO: C to Oracle Enterprise Session Border Controller.	29 - UA-C sends BYE to Oracle Enterprise Session Border Controller.
13 - Oracle Enterprise Session Border Controller forwards REFER-TO: C to UA-B.	30 - Oracle Enterprise Session Border Controller responds with OK to UA-C.
14 - UA-B responds with 202 ACCEPTED to Oracle Enterprise Session Border Controller.	31 - Oracle Enterprise Session Border Controller sends BYE to UA-B.
15 - Oracle Enterprise Session Border Controller forwards 202 ACCEPTED to UA-A.	32 - UA-B responds with OK to Oracle Enterprise Session Border Controller.
16 - UA-B sends INVITE TO: C to Oracle Enterprise Session Border Controller.	33 - Oracle Enterprise Session Border Controller sends BYE to SRS
17 - Oracle Enterprise Session Border Controller sends INVITE to UA-C.	34 - SRS responds with OK to Oracle Enterprise Session Border Controller.

REFER Call (REFER handled by Oracle Enterprise Session Border Controller)

The following illustration shows a call using selective recording and the Session Border Controller (Oracle Enterprise Session Border Controller) handling the REFER on the call. Recording is required in this call flow.

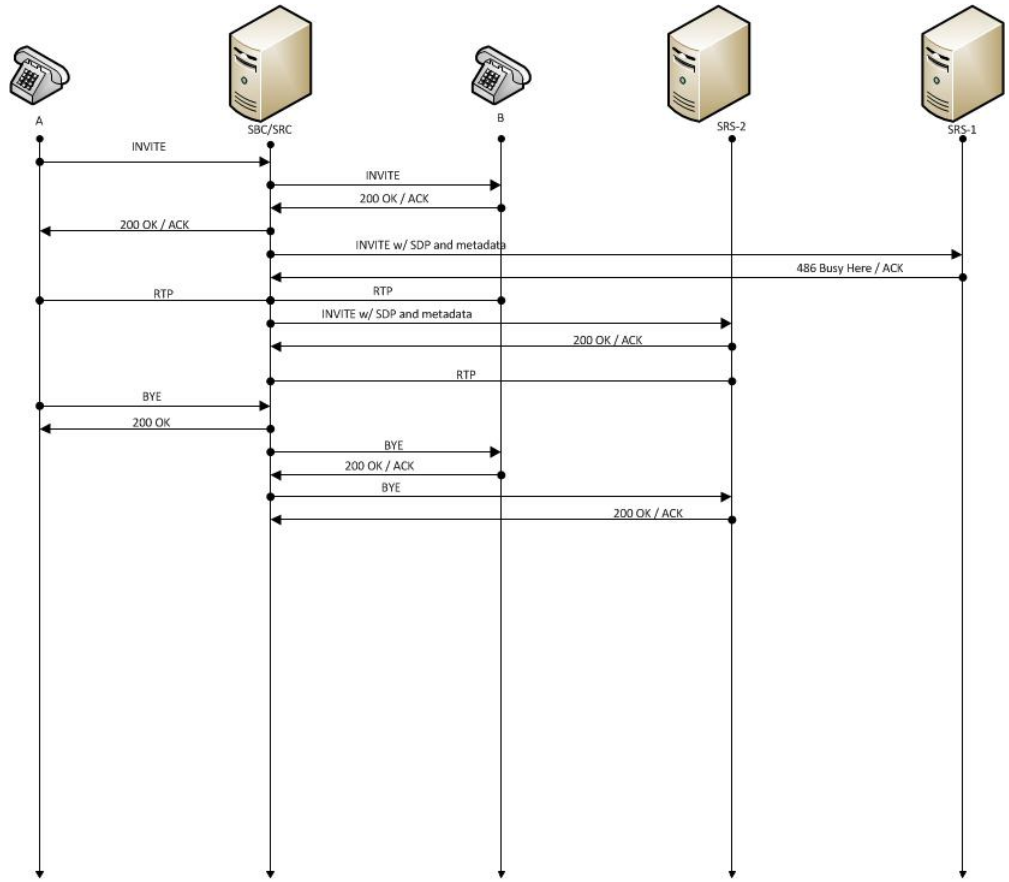


SIP REFER, SBC absorbs REFER, "recording required", selective recording

Call Flow Description	
1 - UA-A sends INVITE to Oracle Enterprise Session Border Controller.	16 - Oracle Enterprise Session Border Controller sends NOTIFY with OK response to UA-A.
2 - Oracle Enterprise Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.	17 - UA-A sends BYE to Oracle Enterprise Session Border Controller.
3 - SRS responds with OK to Oracle Enterprise Session Border Controller.	18 - Oracle Enterprise Session Border Controller responds with OK to UA-A.
4 - Oracle Enterprise Session Border Controller sends INVITE to UA-B.	19 - Oracle Enterprise Session Border Controller sends re-INVITE to UA-B.
5 - UA-B responds with OK to Oracle Enterprise Session Border Controller.	20 - UA-B responds with OK to Oracle Enterprise Session Border Controller.
6 - Oracle Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	21 - Oracle Enterprise Session Border Controller sends re-INVITE to SRS with new SDP and metadata.
7 - SRS responds with OK to Oracle Enterprise Session Border Controller.	22 - SRS responds with OK to Oracle Enterprise Session Border Controller.
8 - Oracle Enterprise Session Border Controller forwards OK response to UA-A.	23 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.
9 - RTP stream initiated between UA-A and Oracle Enterprise Session Border Controller.	24 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-C.
10 - RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.	25 - RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.
11 - RTP stream initiated between Oracle Enterprise Session Border Controller and SRS.	26 - UA-C sends BYE to Oracle Enterprise Session Border Controller.
12 - UA-A sends REFER-TO: C to Oracle Enterprise Session Border Controller.	27 - Oracle Enterprise Session Border Controller responds with OK to UA-C.
13 - Oracle Enterprise Session Border ControllerOracle Enterprise Session Border Controller responds with 202 ACCEPTED to UA-A.	28 - Oracle Enterprise Session Border Controller sends BYE to UA-B.
14 - Oracle Enterprise Session Border Controller sends INVITE to UA-C.	29 - UA-B responds with OK to Oracle Enterprise Session Border Controller.
15 - UA-C responds with OK to Oracle Enterprise Session Border Controller.	30 - Oracle Enterprise Session Border Controller sends BYE to SRS.
	31 - SRS responds with OK to Oracle Enterprise Session Border Controller.

SRS Indicates Busy in Call (recording not required)

The following illustration shows the Session Recording Server (SRS) is BUSY for a call session. Recording is not required in this call flow.



SIP INVITE, "recording not required"

Call Flow Description	
① UA-A sends INVITE to Oracle Enterprise Session Border Controller.	⑨ Oracle Enterprise Session Border Controller sends INVITE to SRS2 with SDP and metadata.
② Oracle Enterprise Session Border Controller forwards INVITE to UA-B.	⑩ SRS2 responds with OK to Oracle Enterprise Session Border Controller.
③ UA-B responds with OK to Oracle Enterprise Session Border Controller.	⑪ RTP stream initiated between Oracle Enterprise Session Border Controller and SRS2.
④ Oracle Enterprise Session Border Controller forwards OK response to UA-A.	⑫ UA-A sends BYE to Oracle Enterprise Session Border Controller.
⑤ Oracle Enterprise Session Border Controller sends INVITE to SRS1 with SDP and metadata.	⑬ Oracle Enterprise Session Border Controller responds with OK to UA-A.
⑥ SRS1 responds to Oracle Enterprise Session Border Controller with 436 BUSY HERE.	⑭ Oracle Enterprise Session Border Controller sends BYE to UA-B.
⑦ RTP stream initiated between UA-A andOracle Enterprise Session Border Controller.	⑮ UA-B responds with OK to Oracle Enterprise Session Border Controller.
⑧ RTP stream initiated between Oracle Enterprise Session Border Controller and UA-B.	⑯ Oracle Enterprise Session Border Controller sends BYE to SRS2.
	⑰ SRS2 responds with OK to Oracle Enterprise Session Border Controller.

Local Media Playback

Commonly, ringback is the media playback of a certain tone informing callers their calls are in progress. In typical deployments, remote endpoints or media servers handle ringback generation, leaving the Oracle Enterprise Session Border Controller to proxy RTP. When endpoints or media servers do not support ringback generation, the Oracle Enterprise Session Border Controller becomes responsible for producing it.

Note: The Oracle Enterprise Session Border Controller supports a maximum of 100 simultaneous playbacks.

You can configure the Oracle Enterprise Session Border Controller to generate ringback locally, meaning it can produce RTP media on a media flow. The most common use for enabling the system to produce RTP on a media flow is to support locally generated ringback. Since you can also use this capability for music-on-hold, announcements, and interrupting media for notifications, this Oracle Enterprise Session Border Controller capability is referred to as local playback.

Local playback is controlled through the ACLI using the Local Media Playback SPL configuration. For more information about SPLs, and configuring the Local Media Playback SPL Plug-in, see Chapter 23, *Session Plug-in Language (SPL)*.

Supported Capabilities and Caveats

The Oracle Enterprise Session Border Controller supports the following playback scenarios:

1. Playback on 183 Session Progress
2. Playback on REFER
3. Playback on header, where the header is P-Acme-Playback

Local media playback is not supported for these Oracle Enterprise Session Border Controller capabilities:

- SRTP
- Call recording
- SIPREC

Local playback does not work in call flows for which media is released. Concurrent playbacks are limited to 100.

Media Setup & Playback

For each session requiring media playback, the Oracle Enterprise Session Border Controller sets up media that supports standard RTP parameters. From the original media (if present), the
Oracle Enterprise Session Border Controller preserves the synchronization source (SSRC), timestamp, and sequence number.

For all playback options besides playback-on-header, the playback duration is continuous, meaning that the media file loops if it fails to cover the entire playback period.

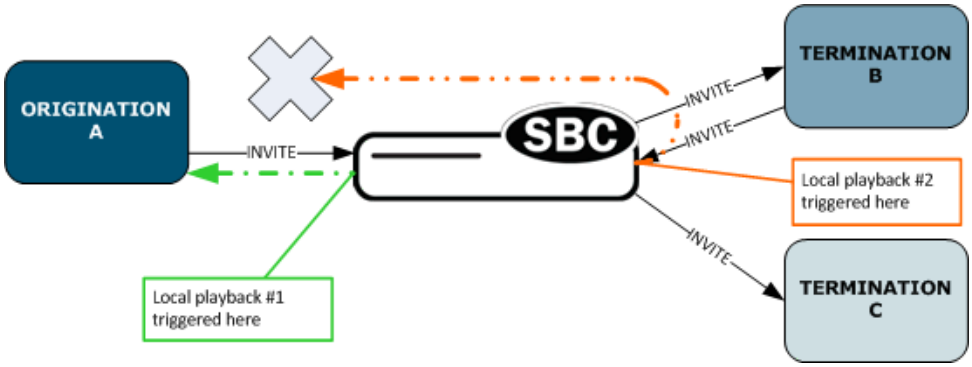
The playback duration for the playback on header scenario changes according to the settings in the P-Acme-Playback header. This header contains a duration=<ms-value|once|continuous> value, so you can customize the playback duration.

- Continuous—Media playback continues until the point at which the playback scenario defines its stop; media file loops if it fails to cover the entire playback duration.
- Once—The file plays until either the playback scenario defines its stop or until the media file runs out.
- Ms-value—Playback continues for a specific duration (in milliseconds) and will loop if necessary.

Once playback is in progress, the Oracle Enterprise Session Border Controller mutes the session in the playback direction so that only the playback media can be heard.

Media Spirals

Certain call flows cause media to traverse the Oracle Enterprise Session Border Controller multiple times, resulting in media spirals. For local playback, this means that multiple playback files can be triggered to play. In situations like this, the Oracle Enterprise Session Border Controller uses the playback closest to the endpoint receiving the media playback. Origination A in the diagram below is played Local playback #1, even though the scenario also triggers Local playback #2.



Supported Playback Scenarios

This section discusses playback scenarios the Oracle Enterprise Session Border Controller supports and identifies triggers for playback. When more than one trigger appears, the Oracle Enterprise Session Border Controller acts on the one closest to the playback endpoint.

These scenarios are defined by the SPL options parameters you configure for realms, session agents, and SIP interfaces. For more information about configuring these options, see Chapter 23, *Session Plug-in Language (SPL)*.

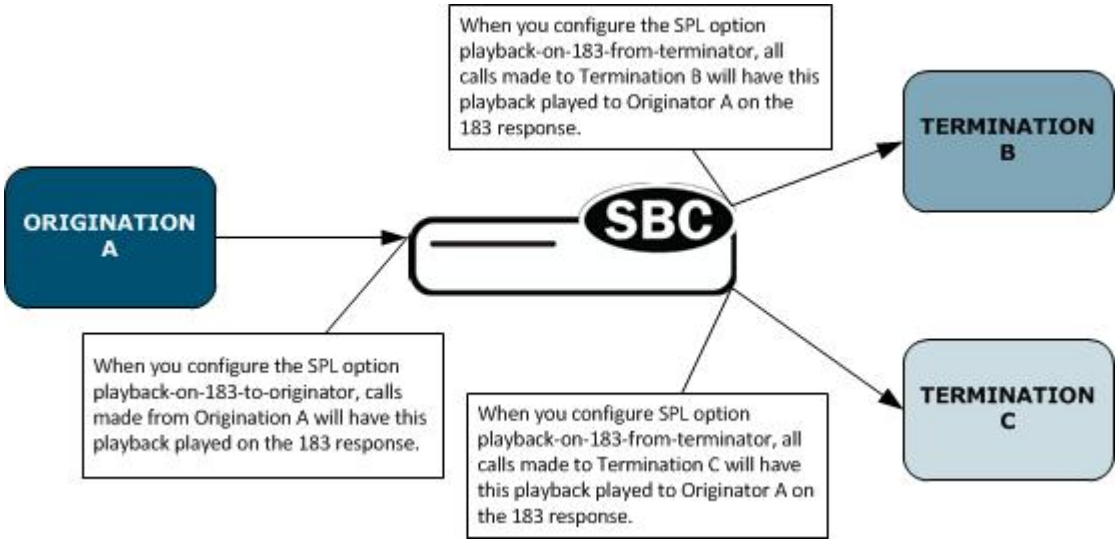
- playback-on-183-to-originator—Playback enabled upon the receipt of a 183 Session Progress destined for the originator and stops when a either a (200-299 or 400-699) final response is sent.
- playback-on-183-from-terminator—Playback enabled upon the receipt of a 183 Session Progress response is received from the terminator and stops when a (200-299 or 400-699) final response is received.
- playback-on-refer—Playback enabled for the caller being transferred when the Oracle Enterprise Session Border Controller receives a REFER message that is locally terminated (i.e., processed on the Oracle Enterprise Session Border Controller on REFER completion).
- playback-on-header—Starts or stops playback based on the presence of the P-Acme-Playback header and its definitions.

Note: The Oracle Enterprise Session Border Controller supports a maximum of 100 simultaneous playbacks.

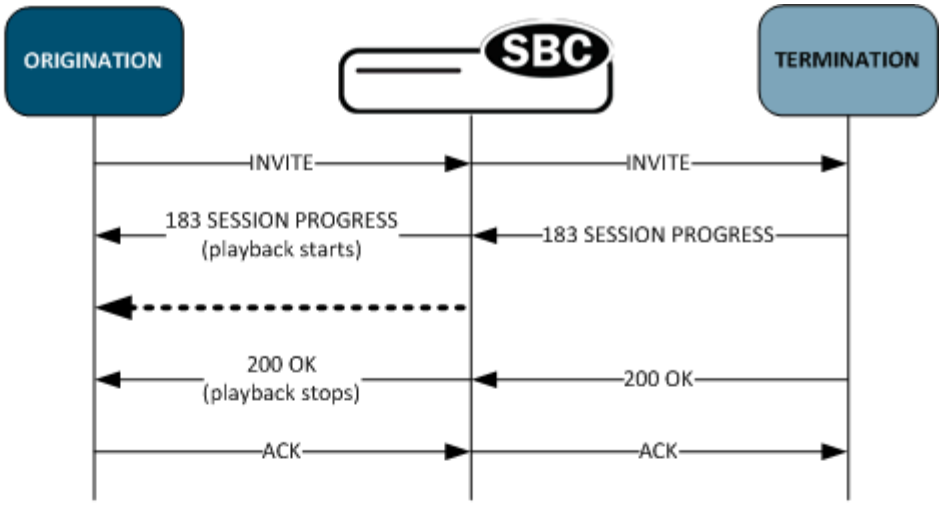
Playback on 183 Session Progress

This scenario is triggered by setting the SPL options parameter to either playback-on-183-to-originator or playback-on-183-from-terminator in realms, session agents, or SIP interfaces. When both options trigger, playback-on-183-to-originator takes precedence. This scenario triggers only for 183 Session Progress responses to initial INVITEs, not for re-INVITEs.

- playback-on-183-to-originator—Starts playback upon the receipt of a 183 Session Progress destined for the originator and stops when a either a (200-299 or 400-699) final response is sent. When you configure this option, every call sent from the originator triggers this playback.
- playback-on-183-from-terminator—Starts playback upon the receipt of a 183 Session Progress response is received from the terminator and stops when a (200-299 or 400-699) final response is received. When you configure this option, every call sent to the terminator triggers this playback.



A call flow for the playback-on-183-from-terminator scenario looks like this:

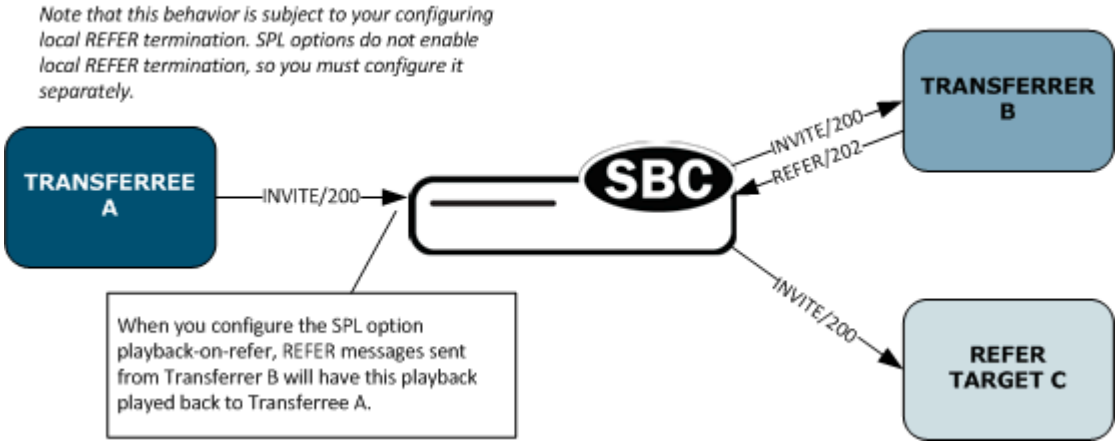

ORIGINATION
SBC
TERMINATION
INVITE
INVITE
183 SESSION PROGRESS
(playback starts)
183 SESSION PROGRESS
200 OK
(playback stops)
200 OK
ACK
ACK

REFER

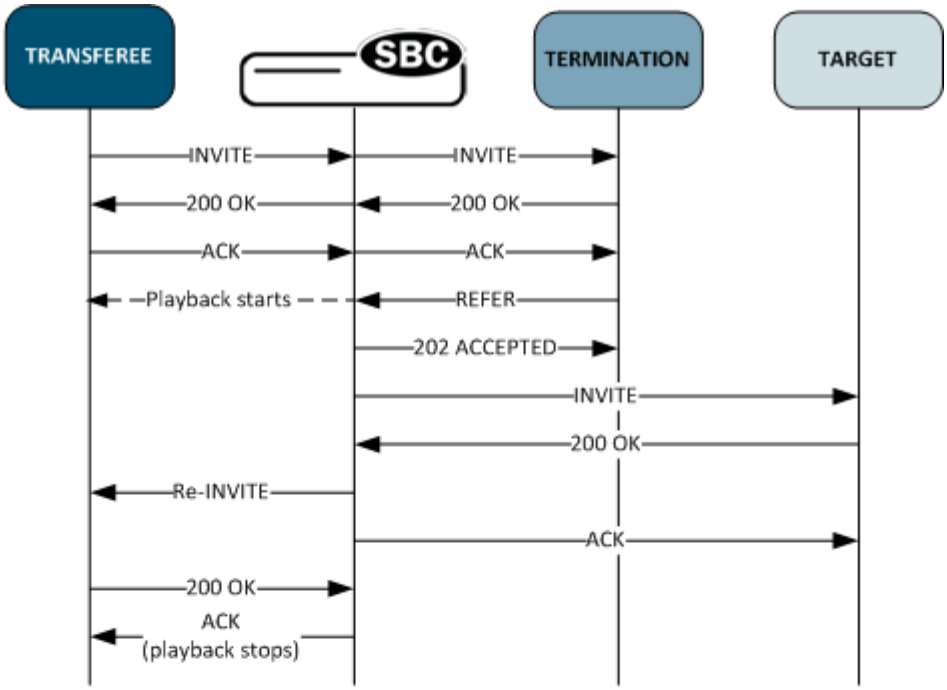
Setting the SPL options parameter to playback-on-refer enables a REFER message to trigger playback. You configure this option for the realm, session agent, or SIP interface for the transferrer, not for the transferee or the REFER target.

The REFER scenario requires that the Oracle Enterprise Session Border Controller performs local REFER termination, i.e., that it terminates the REFER locally. The SPL options you configure do not implement this behavior: You must configure local REFER termination separately. Proxying a REFER message is not a trigger.

Playback begins when the Oracle Enterprise Session Border Controller receives the REFER message, and stops when the REFER operation is deemed complete with a final response (200-299 or 400-699).



A call flow for the playback-on-refer scenario looks like this:

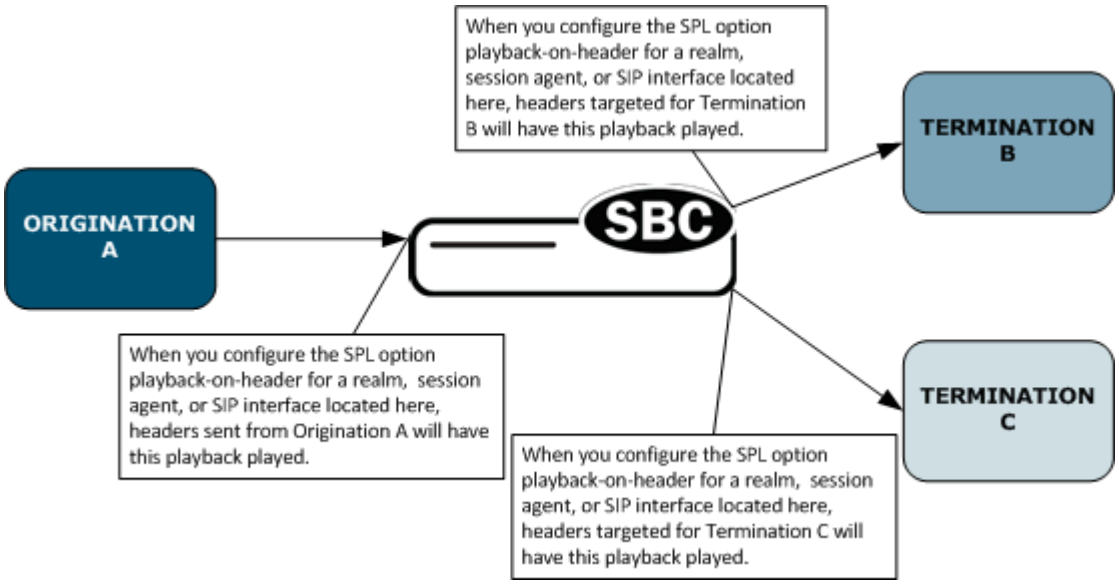

TRANSFEREE
SBC
TERMINATION
TARGET
INVITE
INVITE
200 OK
200 OK
ACK
ACK
Playback starts
REFER
202 ACCEPTED
INVITE
200 OK
Re-INVITE
ACK
200 OK
ACK
(playback stops)

Playback Header

Setting the SPL options parameter to playback-on-header triggers in the presence of the P-Acme-Playback header. You can configure the option on either the side receiving the header message or the side from which the message will be sent. If both trigger, then the configuration closest to the playback direction takes precedence.

This header can be part of any request or response, but playback can only start once media has been established. Playback stops automatically with a final response (200-299 or 400-699), unless explicitly turned off or another playback header requesting it to stop is received.

The Oracle Enterprise Session Border Controller deletes the P-Acme-Playback after processing if the SPL option is configured for the call (either incoming or outgoing).



The header looks like this:

```
P-Acme-Playback: start
    ;media=media1
    ;duration=continuous
    ;direction=both
    ;stop-on-final-resp=true
```

Header Element	Description
<start|stop>	Required Defines starting and stopping playback. • start: starts playback • stop: stops playback
[;media=<media-name>]	Optional Defines the name of the playback configuration to play. If unspecified, the playback configuration that was triggered by the header will play.
[;duration=<ms-value|once|continuous>]	Optional Defines the duration of playback. If unspecified, the value will be taken from the playback-config that was triggered. • ms-value: time value in milliseconds • once: plays playback media one time • continuous: loops the playback media

Header Element	Description
[;direction=<originator|terminator|both>]	Optional Defines the direction from which to play media. If unspecified, playback will begin in the realm, session agent, or SIP interface from which the header was received. • originator: plays in the west flow (original caller) • terminator: plays in the east flow (original callee) • both: plays in both directions
[;stop-on-final-resp=<true|false>]	Optional Defines whether or not to stop playing media upon the final response. If unspecified, this parameter is true. • true: stops playback automatically on a final response • false: stop only after a stop header is received or media terminated

ACLI Configuration and Examples

For configuring the Local Media Playback SPL options on the Oracle Enterprise Session Border Controller, see Chapter 23, *Session Plug-in Language (SPL) (1213)*.

Considerations for HA Nodes

On switchover, media set-up for playback is preserved, which requires negotiated codec and ptime for playback be transferred to the stand-by system in an HA node. However, any playback in progress will not be continued on switchover.

While standard configuration replication handles transferring configuration information between the active and standby systems, media playback files (in /code/media) must be loaded onto the standby.

Alarms

These are the alarms for local playback. They are MAJOR in severity, and do not impact the system health score.

Alarm	Description
Playback media file not found or couldn't be loaded	Raised when a configuration is activated if the system cannot find a media file referenced configuration or if the system is unable to load the media file. This alarm clears automatically when a file is correctly referenced or when it is loaded properly. You might encounter this alarm if you have established playback configuration, but have not loaded the appropriate playback files to /code/media.
Playback could not be started due to capacity limit	Raised at call time when system has reached its maximum number of playbacks (100). This alarm must be cleared manually.
Playback could not be started due to unsupported codec	Raised at call time when there is a mismatch of codecs between those in available files and one that must be played. This alarm must be cleared manually.

Monitoring

You can use the show mbcd statistics command to displays the number of media playbacks that are currently alive:

```
ACMEPACKET# show mbcd statistics
MBCD Status                       -- Period -- -------- Lifetime --------
                        Active    High   Total       Total  PerMax    High
Media Playback               0       5       5           6       0       6
```

You can use the show mbcd errors command to track the number of playback failures:

```
ACMEPACKET# show mbcd errors
MBC Errors/Events               ---- Lifetime ----
                          Recent      Total  PerMax
Media Playback Fails           0          0       0
Playback Exh Resources         0          0       0
Playback Flow Inactive         0          0       0
Playback Mismatch              0          0       0
```

10

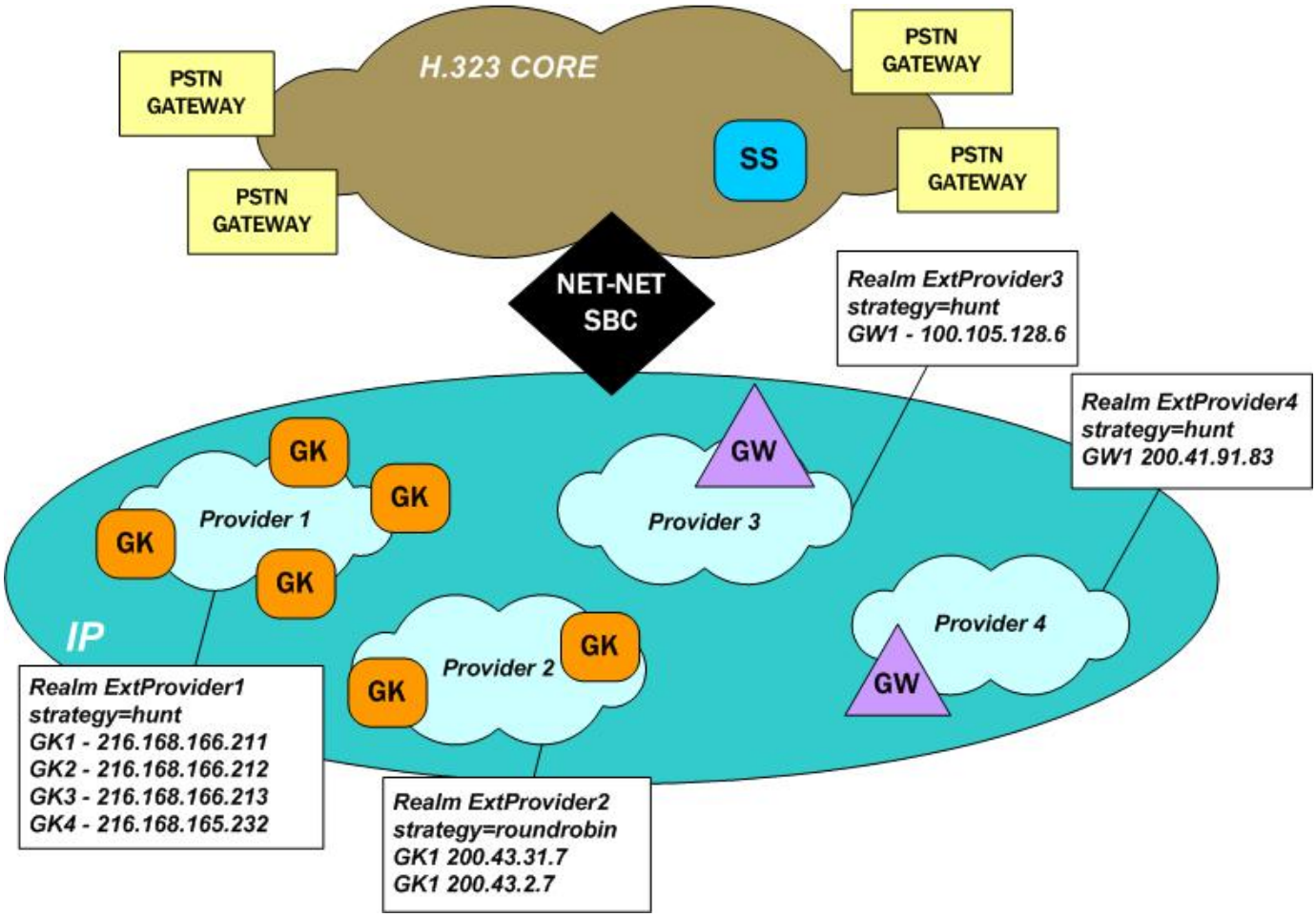
H.323 Signaling Services

The Oracle Enterprise Session Border Controller supports H.323 signaling in a way that permits interworking between different H.323 configurations from different providers and carriers. H.323 signaling capabilities on the Oracle Enterprise Session Border Controller include:

- H.323 V4—Improves on previous versions of the protocol in functionality, scalability, and reliability
- H.225 call signaling with RAS—Establishes connections between H.323 endpoints so real-time data can be exchanged
- H.245—Establishes the type of media flow and manages that flow after it has started
- H.245 tunneling—Encapsulates H.245 messages within H.225/Q.931 messages; when enabled and used with a firewall, one less TCP port is needed for incoming connections
- Fast Start (and Fast Start with parallel H.245)
- H.323 Annex E support for UDP signaling—Provides for multiplexed call signaling over UDP to increase potential call volume and enhance performance

Peering Environment for H.323

The following diagram shows a peering environment for H.323, with the Oracle Enterprise Session Border Controller positioned between the H.323 core and external providers.



The configuration information shown in the diagram can help you to understand how some basic Oracle Enterprise Session Border Controller concepts work. The providers in this depiction are configured as realms, and the strategies you see are for session agent group. What you do not see in this diagram is the fact that the Oracle Enterprise Session Border Controller is configured with sets of H.323 interfaces within it. These interfaces are internal (for an internal provider) and external (for the external providers you see).

Video-Conferencing Support

The Net-Net Session Director (NN-SD) supports H323 video-conferencing environments using the H239 Procotol for video-conferencing. It provides critical control functions to enable high quality interactive communication—voice, video and multimedia sessions—across IP network borders.

For additional information about the H323 Protocol, see the Net-Net 4000 ACLI Configuration Guide, Version 6.3.

The NN-SD architecture supports both voice and video applications. It uses Codec Media Profiles to determine the proper amount of bandwidth allocated for a given session, distinguishing between G.711 or G729 voice and H. 263/264 video requirements. By supporting video transmission as well as voice over the IP Multi-protocol label switching (MPLS) core, the NN-SD allows Service Providers to roll out new services to their enterprise customer such as video/audio conferencing.

Note: IP MPLS is a packet-switched network that uses the Internet Protocol (TCP/IP) enhanced with the Multi-protocol label switching (MPLS) standard.

The NN-SD allows for aggregate bandwidth policies to be configured for each realm. As the NN-SD processes call requests (to and from) a particular realm, the bandwidth consumed for the call is decremented from the bandwidth pool for that realm. The SD determines the required bandwidth from the SDP/H.245 information. Any request that

would cause the bandwidth constraint to be exceeded is rejected with a SIP “503 Service Unavailable” or an H.323 Release Complete.

To alleviate the bandwidth demands of high-definition video streams, the NN-SD offers a 2 or 4 Gigabit PHY card option.

Video Conferencing Support for Polycom Terminals

The Oracle Enterprise Session Border Controller in a video conferencing environment with Polycom H323 terminals and a Polycom MCU (Multipoint Conferencing Unit) relays H.239/H.245. The Oracle Enterprise Session Border Controller implements the following messages appropriately:

- Miscellaneous command message with subtype such as multiPointModeCommand, cancelMultipointModeCommand
- Conference Indication message with subtype such as terminalNumberAssign, terminalYouAreSeeing

Overview

Using H.323 on your Oracle Enterprise Session Border Controller, you can implement different signaling modes and use features to enhance H.323 capabilities. In the information that follows, you will find detailed explanations of the H.323 signaling mode and of the features available. This chapter gives operational details and later outlines the steps you need to take when features require configuration.

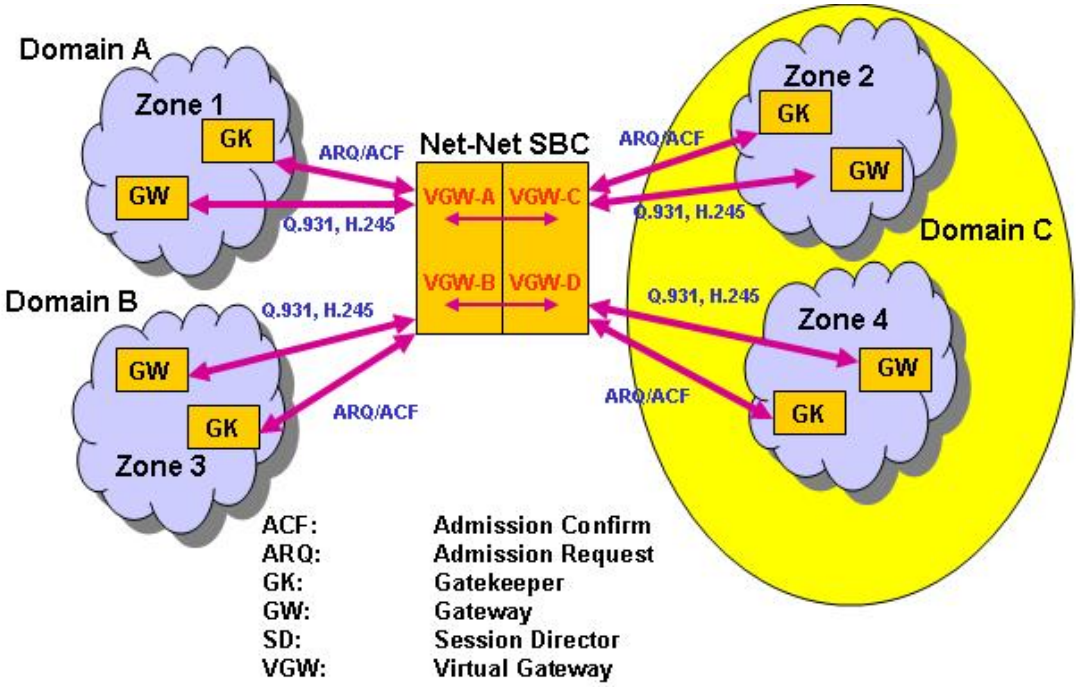
Signaling Modes of Operation

Your Oracle Enterprise Session Border Controller can operate in different H.323 signaling modes:

- Back-to-back gateway signaling
- Back-to-back gatekeeper proxy and gateway
- Interworking gatekeeper/gateway

Back-to-Back Gateway Signaling

This section explains how signaling takes place when the Oracle Enterprise Session Border Controller functions as a B2BGW for H.323. The following diagram illustrates the Oracle Enterprise Session Border Controller acting as a B2BGW.



When configured as a B2BGW, the Oracle Enterprise Session Border Controller appears as multiple H.323 gateways to multiple networks. You can think of the Oracle Enterprise Session Border Controller as having virtual gateways, that discovers and registers with a gatekeeper in its respective domain. In this configuration, you need to set the service mode (isgateway) parameter for the H.323 interface to enabled for two H.323 interfaces. These interfaces are related either through their outgoing interface (assoc-stack) parameters or through routing policies.

If you configure your Oracle Enterprise Session Border Controller to operate in this mode, it does not issue or respond to LRQs by either confirming them or rejecting them.

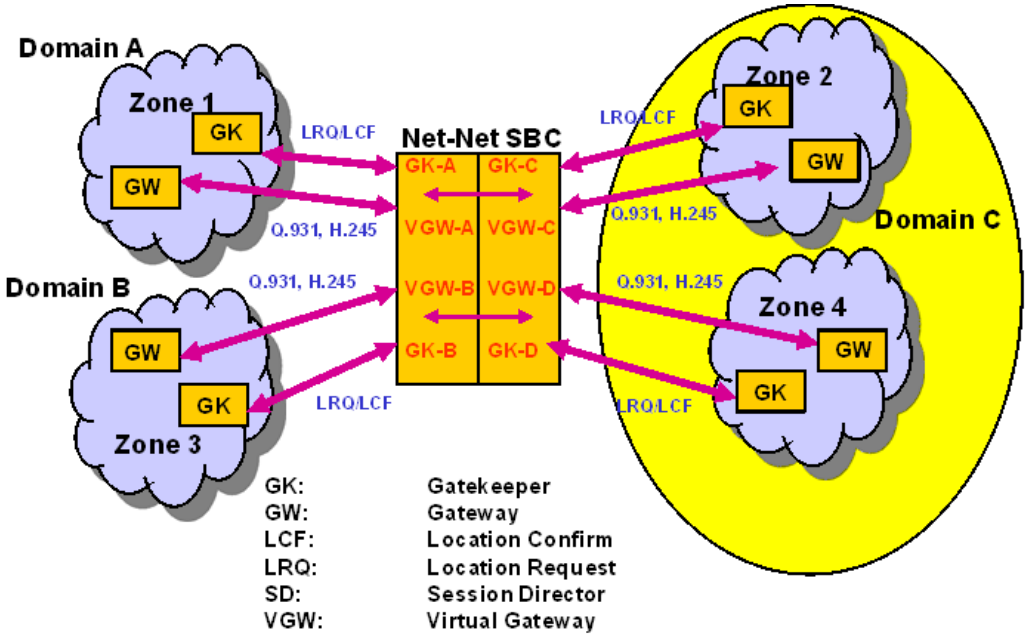
In the diagram above, the Oracle Enterprise Session Border Controller sends ARQs to the corresponding gatekeeper in its zone when a call is received on the associated interface. In this behavior, the Oracle Enterprise Session Border Controller acts as a gateway, complying with the H.323 standard, and registers with the configured gatekeeper in its assigned zone. You set all parameters related to the gateway registrations, such as gateway prefix numbers, in the H. 323 interface configuration.

In this mode, you can also configure the Oracle Enterprise Session Border Controller to run like a gateway without a gatekeeper by turning off automatic discovery (auto-gk-discovery) for the remote gatekeeper. When the Oracle Enterprise Session Border Controller receives a Setup message, it does not send an ARQ and there is no registration for admission requests. Without automatic gateway discovery, the Oracle Enterprise Session Border Controller uses the local policy to find the appropriate destination for the call. This destination is normally the IPv4 address of the endpoint or gateway, using the well-known port 1720.

If you enable this capability, then the Oracle Enterprise Session Border Controller finds a gatekeeper.

Back-to-Back Gatekeeper Proxy and Gateway

This section explains how signaling takes place when the Oracle Enterprise Session Border Controller functions as a back-to-back gatekeeper proxy and gateway for H.323. The following diagram illustrates theOracle Enterprise Session Border Controller acting as a B2B gatekeeper proxy and gateway.



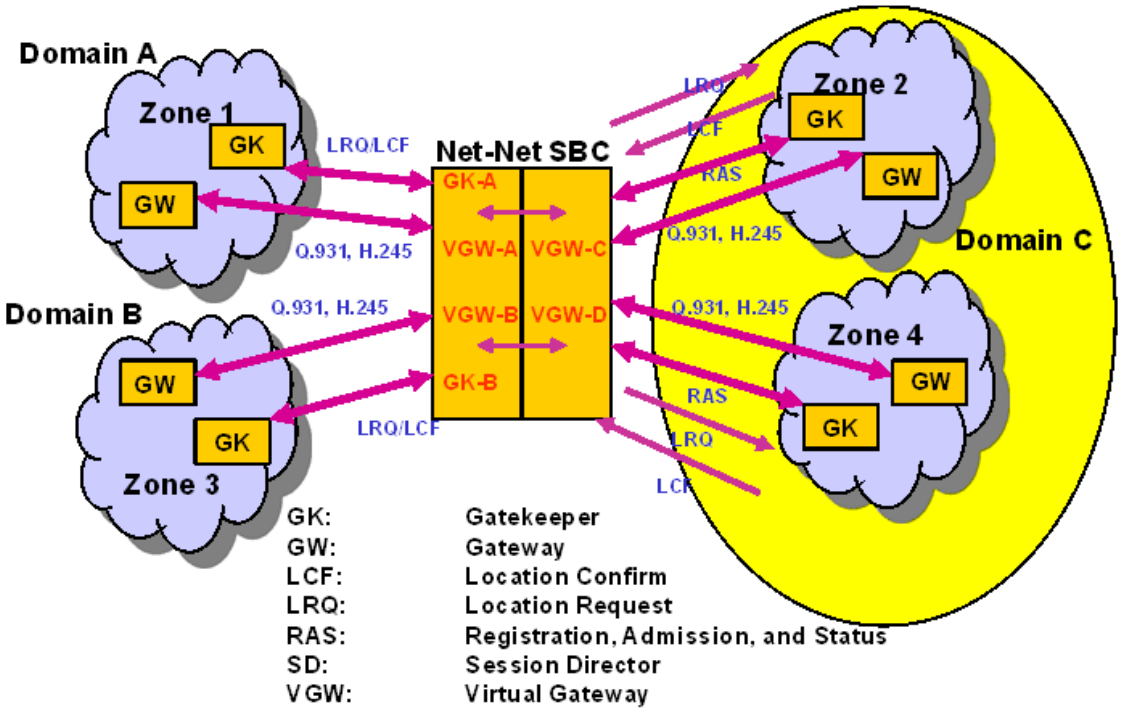
In this application, with the service mode (isgateway) parameter set to disabled, the Oracle Enterprise Session Border Controller responds to LRQs and issues LCFs and LRJs. It sends LRQs and LCFs/LRJs to the local IPv4 address for the H.323 interface. The Oracle Enterprise Session Border Controller responds to the LRQs by providing a signaling address that performs gateway functions.

When you use it as a back-to-back gatekeeper proxy and gateway, the Oracle Enterprise Session Border Controller does not issue ARQs. In addition, all parameters related to registration, such as gateway prefix numbers, are ignored.

When you do not configure a gatekeeper, the Oracle Enterprise Session Border Controller uses the local policy to find the appropriate destination for the call. If there is a matching local policy, the Oracle Enterprise Session Border Controller returns an LCF to the originating gateway. If no local policy matches, the Oracle Enterprise Session Border Controller rejects the call by sending an LRJ.

Interworking Gatekeeper-Gateway

This section explains how signaling takes place when the Oracle Enterprise Session Border Controller functions as an interworking gatekeeper-gateway for H.323. The following diagram shows the Oracle Enterprise Session Border Controller acting as an interworking gatekeeper-gateway.



When you configure your Oracle Enterprise Session Border Controller for interworking gatekeeper-gateway mode, one H.323 interface behaves as a B2BGW and its associated interface for the corresponding network behaves like a gatekeeper proxy and gateway. The interface for the gatekeeper proxy and gateway issues and responds to LRQ messages on its network. If the Oracle Enterprise Session Border Controller knows the gatekeeper in the network of the gateway interface (Zone 2), it sends an LRQ to that gatekeeper. If the gatekeeper responds with an LCF or LRJ, the Oracle Enterprise Session Border Controller forwards it.

If the gatekeeper (in Zone 2) is unknown, then the Oracle Enterprise Session Border Controller responds to LRQs on the gatekeeper-gateway network (Zone 1) by using the local policy to determine the appropriate destination for the LRQ. If there is no local policy that matches, then the Oracle Enterprise Session Border Controller sends an LRJ.

For this configuration, the gateway interface has its service mode (isgateway) set to enabled, and the gatekeeper interface has its service mode (isgateway) set to disabled.

Realm Bridging with Static and Dynamic Routing

The Oracle Enterprise Session Border Controller uses static routing and policy-based, dynamic routing to handle H.323 traffic. These types of routing have to do with the way that the outgoing stack is selected.

- Static routing—The incoming H.323 stack always uses the associated H.323 stack that you configure for outgoing traffic; no other stacks are considered.
- Dynamic routing—When there is not an associated stack configured, the Oracle Enterprise Session Border Controller performs policy-based, dynamic routing known as realm bridging. In this type of realm bridging, the Oracle Enterprise Session Border Controller checks the configured local policies for address information corresponding to the incoming traffic and finds an address that matches. Next, it checks the next hop in the local policy to determine a realm and uses the first H.323 interface that matches it.

Before You Configure

In order to run H.323 on your Oracle Enterprise Session Border Controller, you need to configure the basic parameters: physical and network interfaces; global system parameters; SNMP, trap receiver, and accounting support, and any holiday information you might want to set.

You should also decide how you want to set up realms and routing (including the use of session agents and session agent groups) to support H.323 operations.

Global H.323 Settings

When you configure H.323 signaling for your Oracle Enterprise Session Border Controller, you set global and per-interface parameters. The global parameters govern how the Oracle Enterprise Session Border Controller carries out general H.323 operations, and these settings are applied to all interfaces you configure for H.323 use. For example, you can turn H.323 support on and off for the entire Oracle Enterprise Session Border Controller using these settings.

Global H.232 Settings Configuration

For the ACLI, global H.323 parameters are:

```
state          State of the H.323 protocol
log-level      Log level for H.323 stacks
response-tmo   maximum waiting time in sec for response to a SETUP message
connect-tmo    maximum waiting time in sec for establishment of a call
options        optional features/parameters
```

Accessing Global H.323 Parameters

To access the global H.323 configuration parameters in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# h323
   ```

   From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt. Access to the H.323 interface (h323-stack) configuration also appears.

Global H.323 Settings

To configure global H.323 parameters:

1. state—Enable or disable the state of H.323 signaling. The default value is enabled. Valid values are:
   - enabled | disabled
2. response-tmo—Enter the amount of time in seconds that the Oracle Enterprise Session Border Controller waits between sending a Setup message and tearing it down if there is no response. The default value is 4 and we recommend you leave this parameter set to this value. The valid range is:
   - Minimum—0
   - Maximum—999999999

   A response might be any of the following messages: Call Proceeding, Connect, or Alerting.
3. connect-tmo—Enter the amount of time in seconds that the Oracle Enterprise Session Border Controller waits between sending a Setup message and tearing it down if it does not specifically receive a Connect message from the endpoint. The default is 32 and we recommend that you leave this parameter set to this value. The valid range is:
   - Minimum—0
   - Maximum—999999999

   Receiving a Proceeding or Alert message from the endpoint does not keep this timer from expiring.

4. options—Set any options for H.323 features that you want to use. This parameter has a global impact on H.323 behavior, rather that being applied on a per-interface basis.

   If you do not configure options for global H.323 behavior, none appears in the configuration display.

5. log-level—Set the process log level for monitoring all H.323 activity on the Oracle Enterprise Session Border Controller. The default is INFO and leaving this parameter set to this value provides an intermediate amount of detail in the logs. Other valid values are:

   Note: Any log level you set here overrides the log level you set in the system configuration's process log level parameter.

Numerical Code	Acme Packet Log Enumeration	Description
1	EMERGENCY	Logs conditions of the utmost severity that require immediate attention.
2	CRITICAL	Logs events of serious condition that require attention as soon as possible.
3	MAJOR	Logs conditions indicating that functionality is seriously compromised.
4	MINOR	Logs conditions indicating that functionality has been impaired in a minor way.
5	WARNING	Logs conditions indicating irregularities in performance.
6 7 8 9	NOTICE INFO TRACE DEBUG	For Acme Packet customer support.

H.323 Interfaces

You need to configure H.323 interfaces for inbound and outbound traffic. When you configure H.323 interfaces, you can set:

- Identity and state
- Realm and H.323 interface associations
- H.323 interface settings for the interface's IPv4 address, RAS and Q. 931 ports, maximum number of Q.931 ports to allow, and any Annex E support you need
- H.323 system resource allocation

H.232 Interfaces Configuration

These are the ACLI parameters that you set:

```
name                  Name of the stack
state                 State of the stack
isgateway             Enable the stack to run as a gateway
terminal-alias        List of aliases for terminal
ras-port              Listening port for RAS request
gk-identifier         Gatekeeper's identifier
q931-port             Q.931 call signalling port
alternate-transport   Alternate transport addresses/ports
q931-max-calls        Maximum number of Q.931 calls
```

```
max-calls              Stack's maximum number of calls
max-channels           Maximum number of channels per channel
```

To access the H.323 interface (h323-stack) and service mode parameters:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type h323-stacks and press Enter.

   ```
   ACMEPACKET(h323)# h323-stacks
   ACMEPACKET(h323-stacks)#
   ```

   From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a ? at the system prompt. The display also includes H.323 service mode parameters.

Identity and State

To set the identity and state of the H.323 interface:

1. name—Enter a name for the H.323 interface using any combination of characters entered without spaces. For example: InternalGK1.
2. state—Enter the state of this H.323 interface. The default value is enabled. Valid values are:
   - enabled | disabled

Realm and Interface Associations

To link this H.323 interface to a realm and to an outgoing H.323 interface:

1. realm-id—Enter the identifier for the realm served by this H.323 interface. This parameter must be configured with a valid identifier value from a realm configuration.
2. assoc-stack—Enter the name of the outgoing H.323 interface that you want to associate with the H.323 interface you are configuring. To use realm bridging with static routing, you need to set the outgoing H.323 interface. If you do not enter a name, the Oracle Enterprise Session Border Controller uses dynamic, policy-based selection using the local policy.

H.323 Signaling Interface Settings

You can set the following parameters to define basic settings for your H.323 interface. This is where you set the IPv4 address for opening sockets, the RAS and Q.931 ports, and the maximum number of Q.931 calls that you want to allow.

This is also where you establish Annex E alternate transport. Annex E supports multiplexed call signaling over UDP so that call volume and performance are potentially enhanced. If you do not configure Annex E support, then this H. 323 interface does not listen for Annex E requests.

To configure H.323 interface settings:

1. local-ip—Enter the IPv4 address that the H.323 interface uses when opening sockets; this is the default H.323 interface IPv4 address. You must use a valid IPv4 address. For example: 192.168.2.5. The default value is 0.0.0.0.
2. ras-port—Enter the number of the port on the local IPv4 address (local-ip) on which the Oracle Enterprise Session Border Controller listens for RAS requests. We recommend that you set this parameter to its default, the well-known port 1719. The valid range is:
   - Minimum—0
   - Maximum—65535

If you set this parameter to 0, the Oracle Enterprise Session Border Controller uses a port assigned by the operating system.

3. q931-port—Enter the number for the port on the local IP address for the Q.931 call signaling port. We recommend that you leave this parameter set to its default, 1720. The valid range is:
   - Minimum—0
   - Maximum—65535
4. q931-max-calls—Enter the maximum number of concurrent Q.931 calls you want to allow. The default value is 200; however, this value should be less than the maximum number of calls you set when configuring H.323 features. The valid range is:
   - Minimum—0
   - Maximum—65535

   If the number of received Q.931 calls exceeds this number, the H.323 interface returns a busy state.
5. alternate-transport—Enter a list of one or more Annex E IPv4 address and port combinations for alternate transport. If you do not configure this list, then the Oracle Enterprise Session Border Controller does not listen for incoming Annex E requests. You must enter the IPv4 address and port combination in the following format, where the two are separated by a colon: IPv4Address:Port.

H. 323 System Resource Allocation

You can set the following parameters to determine how many concurrent calls and concurrent channels you want to allow for each H.323 interface.

To allocate H.323 system resources:

1. max-calls—Enter the maximum number of concurrent calls allowed on this H.323 interface. The default value is 200. The valid range is:
   - Minimum—0
   - Maximum—4294967295
2. max-channels—Enter the maximum number of concurrent channels allowed for each call associated with this H. 323 interface. The default value is 6. The valid range is:
   - Minimum—0
   - Maximum—4294967295

   The Oracle Enterprise Session Border Controller checks this parameter on initialization to reserve the appropriate network resources.

H.323 Service Modes

When you set the H.323 service mode, you configure parameters that define what type of service an H.323 interface provides. These parameters govern how the interface functions when you want it to behave as a gatekeeper or as a gateway.

This is also where you set options that support particular H.323 features for a specific interface. These options are different from the ones you set in the global H.323 configuration because they apply only to the interface where you specify them.

H.232 Service Modes Configuration

These are the ACLI parameters that you set:

```
isgateway              Enable the stack to run as a gateway
registration-ttl       Number of seconds before the registration becomes
invalid
terminal-alias         List of aliases for terminal
auto-gk-discovery      Enable automatic gatekeeper discovery
```

```
multicast             RAS multicast address
gatekeeper            Gatekeeper's address and port
gk-identifier         Gatekeeper's identifier
h245-tunneling        Enable H.245 Tunneling support
prefixes              List of supported prefixes
process-registration  Enable Registration Request processing
allow-anonymous       allowed requests from H.323 realm
```

To configure the service mode for the H.323 interface:

1. allow-anonymous—Enter the admission control of anonymous connections from an H.323 realm accepted and processed by this H.323 stack. The default value is all. The valid values are:
   - all—Allow all anonymous connections
   - agents-only—Allow requests from session agents only
   - realm-prefix—Allow session agents and addresses matching the realm prefix
2. is-gateway—To use this interface as an H.323 gateway, leave this parameter set to enabled, its default value. If you want to use this interface as an H.323 gatekeeper, set this parameter to disabled. Valid values are:
   - enabled | disabled
3. terminal-alias—Enter a list of one or more aliases that identify the H.323 interface. This value is either the gateway alias or the gatekeeper identifier, depending on the mode you configure for the interface. The aliases are set in the sourceInfo information element of outgoing ARQs.

Configuring Gateway Only Settings

If you are using the H.323 interface as a gateway, you might want to set registration time-out and address prefix parameters.

To configure gateway only settings:

1. registration-ttl—Enter the number of seconds before a registration becomes invalid. This value is used during the initial registration process. However, when a registration is confirmed, the time-to-live (TTL) value set by the gatekeeper in the Registration Confirm (RCF) message overrides this value. The default value is 120. The valid range is:
   - Minimum—0
   - Maximum—4294967295
2. prefixes—Enter a list of prefixes for this H.323 interface. Possible prefix types include:
   - H.323 ID | E.164 | URL | IPv4 address

     These prefixes are sent from a gateway interface to a gatekeeper and indicate valid prefixes accepted by that interface for incoming calls. They are used if the interface is configured as a gateway (the is-gateway parameter is set to enabled).

     Your entries for this parameter must appear as they do in the following example:

     ```
     e164=17817566800 url=http://www.companyname.com
     h323-ID=xyz email=user@companyname.com
     ipAddress=63.67.143.4:2000
     ```

Gatekeeper Proxy Settings

If you are using the H.323 stack as a gatekeeper proxy, you might want to set:

- Whether registration processing is enabled or disabled
- Whether or not this H.323 interface is signaling-only
- At what H.225 call stage the H.245 procedures should be initiated

To configure gatekeeper proxy settings:

1. process-registration—To have the Oracle Enterprise Session Border Controller drop all RRQs, meaning that it does not acknowledge any requests, leave this parameter set to disabled, its default. To have the Oracle Enterprise

Session Border Controller process any RRQs that arrive on this H.323 interface, set this parameter to enabled. Valid values are:

- enabled | disabled

  When registration processing is enabled and the Oracle Enterprise Session Border Controllerreceives an RRQ on this H.323 interface, it will route the request to the appropriate gatekeeper. After the gatekeeper confirms that registration with an RCF, the Oracle Enterprise Session Border Controller also confirms it with the endpoint that sent the RRQ. Then the registration becomes part of the Oracle Enterprise Session Border Controller's registration cache. If this endpoint does not confirm the registration, then the Oracle Enterprise Session Border Controller will reject the registration with an RRJ and will not cache it.

2. proxy-mode—Set this field to the proxy mode that you want to use for the signaling only operation mode. Valid values are:

   - H.225 | H.245

     You can leave this field blank (default) if you are not using a proxy mode.

3. h245-stage—Set this field to the stage at which the Oracle Enterprise Session Border Controller transfers the H. 245 address to the remote side of the call, or acts on the H.245 address sent by the remote side. The default value is connect. Valid values are:

   - Setup | Alerting | Connect | Proceeding | Early | Facility | noh245 | Dynamic

H.323 Features

This section provides general descriptions of the H.323 features available on the Oracle Enterprise Session Border Controller and instructs you in how to configure them. Not all of the features described in that chapter require configuration.

Fast Start Slow Start Translations

The Oracle Enterprise Session Border Controller can translate between Fast Start H.323 endpoints and Slow Start H. 323 endpoints. Using this feature, you can reduce delay in establishing media, improve performance, and reduce network congestion caused by a high number of messages being exchanged. Fast Start and Slow Start calls handle information about media for a session in different ways. In a Fast Start call, information about the media is contained in the Setup message. In a Slow Start call, that information is exchanged between endpoints after the session has been established.

When you Fast Start/Slow Start translation, the Oracle Enterprise Session Border Controller can take a Slow Start call from an H.323 endpoint that does not support Fast Start and re-initiate that call as Fast Start. It also allows an H.323 endpoint that does not support Fast Start to receive a Slow Start call from a Fast Start source because the Oracle Enterprise Session Border Controller performs all necessary translations.

For the ACLI, the following parameters apply:

```
fs-in-first-msg       Fast Start must be sent in 1st response to Setup message
call-start-fast       Enable outgoing Fast Start call
call-start-slow       Enable outgoing Slow Start call
media-profiles        list of default media profiles used for outgoing call
```

Fast Start to Slow Start Translation

The Oracle Enterprise Session Border Controller supports translations from H.323 Fast Start to Slow Start. Using this feature, an H.323 endpoint that only supports Slow Start can call from a Fast Start source when that call goes through the Oracle Enterprise Session Border Controller.

In a Fast Start call, the originating H.323 endpoint sends a fastStart element in its Setup message. This element contains H.245 OLC messages that allow Fast Start endpoints to establish a media stream when the call is connected. As a result fewer messages are exchanged between the H.323 endpoints than there would be for a Slow Start call (where the fastStart element does not appear). Because media information is sent in the Setup request for the session, there is no need to use the media profiles when converting a Fast Start call to Slow Start.

When you enable the slow start option in the H.323 stack configuration, the Oracle Enterprise Session Border Controller performs Fast Start to Slow Start conversion. During the translation, the Oracle Enterprise Session Border Controller retains the media information included in the incoming Fast Start call as it negotiates a connection with the Slow Start endpoint. After a connection with the Slow Start endpoint has been established, the Oracle Enterprise Session Border Controller negotiates the media capabilities.

Slow Start to Fast Start Translation

When you configure your Oracle Enterprise Session Border Controller to support H.323 Slow Start to Fast Start translations, you enable an H.323 endpoint that only supports Slow Start to initiate and sustain communication with an H.323 Fast Start endpoint. The Oracle Enterprise Session Border Controller resolves the Slow Start limitation of exchanging information about media (OLC messages) after the call is connected. The OLC message opens a logical channel, or a unidirectional or bi-directional path used to transmit media packets. Using the Oracle Enterprise Session Border Controller, you can negotiate the construction of media flows differently, which is described in this section.

When you enable the Fast Start option for calls in the H.323 stack configuration, the Oracle Enterprise Session Border Controller performs the translation of a Slow Start call into Fast Start. When it receives a Slow Start call, the Oracle Enterprise Session Border Controller determines its destination and the H.323 stack it uses for the outgoing call.

It is a requirement of this kind of translation that you configure and use media profiles. Since a Slow Start call does not negotiate media until after the call is connected, there needs to be an assumption made about the media to set up a Slow Start to Fast Start call. Media profiles fill this role, and they are assumed to be part of a correct configuration.

The following describes possible scenarios for Slow Start to Fast Start translations.

- When a Slow Start call arrives at the Oracle Enterprise Session Border Controller and matches one of the session agents that has a media profiles list configured, the outgoing call is set up as a Fast Start call. The session agent's media profiles are used for the logical channels. You must configure the media profiles to reference a codec the endpoint accepts.

  If there are no media profiles configured for the session agent, then the Oracle Enterprise Session Border Controller uses the media profiles list in the H.323 stack configuration to open the logical channels.
- If a Slow Start calls arrives at the Oracle Enterprise Session Border Controller and its destination does not match one of the session agents, the Oracle Enterprise Session Border Controller uses the media profiles list in the H.323 stack configuration for the outgoing call. If there is a list of media profiles, the outgoing call is set up as a Fast Start call with the media profiles list used to open the logical channels.

  If there is no list of media profiles for the outgoing H.323 interface, the Oracle Enterprise Session Border Controller does not perform Slow Start to Fast Start translation. The Slow Start call exits the Oracle Enterprise Session Border Controller as it arrived—as a Slow Start call.
- If the egress H.323 interface has the Fast Start option disabled, then the outgoing call uses the Slow Start mode, and the Oracle Enterprise Session Border Controller does not perform Slow Start to Fast Start translation. In this case, the Slow Start call also exits the Oracle Enterprise Session Border Controller as it arrived—as a Slow Start call.

Slow Start Fast Start Prerequisites

To perform Fast Start/Slow Start translations, you need to have a standard two-interface configuration already in place.

If you are using the Slow Start to Fast Start translations, you must configure appropriate entries in the media profiles list which is part of the translation parameters. The Fast STart/Slow Start Translations section of the Oracle Enterprise Session Border Controller Feature chapter describes how the media profiles are used. The list contains the names of media profiles that you configure in the media profile configuration.

Some media profiles are configured by default. If the information you have configured for a media profile collides with the defaults, then your configured ones are loaded. If there are no collisions, then the Oracle Enterprise Session Border Controller loads the configured and default profiles. The default media profiles are:

Type	Payload	Encoding	Bandwidth
audio	0	PCMU	0
audio	2	G726-32	0
audio	4	G723	0
audio	8	PCMA	0
audio	9	G722	0
audio	15	G728	0
audio	18	G729	0
audio	101	telephone-events	0

Ensure that you use the name of a configured media profile when you enter values in the media profiles list.

Media Profile Configuration

In the ACLI, you can set media profiles that are required for translating H.323 Slow Start to Fast Start. In the ACLI, you set the following:

```
name                encoding name used in sdp rtpmap attribute
media-type          media type used in sdp m lines
payload-type        rtp payload type used in sdp m lines
transport           transport protocol used in sdp rtpmap attribute
req-bandwidth       amount of bandwidth in kilobits required
frames-per-packet   maximum number of frames per packet
parameters          list of <name=value> pairs separated by space
average-rate-limit average rate limit of rtp flow
```

To configure a media profile:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type media-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# media-profile
   ```

   From this point, you can configure media profiles parameters. To view all media profiles configuration parameters, enter a ? at the system prompt.

4. name—Enter the encoding name used in the SDP rtpmap attribute. You must enter a name to uniquely identify the media profile, and you will use this value to make lists of media profiles in H.323 interface configurations.
5. media-type—Leave this parameter set to its default, audio. Valid values are:

   - audio | video | application | data | image | text

6. payload-type—Enter the payload type number that corresponds to the encoding name you entered in Step 4. This value identifies the format in the SDP m lines. There is no default value for this parameter. The About Payload Types section contains a table of standard audio and visual encodings.

   Note: When you use the RTP/AVP transport method, this value must be numeric.

7. transport—Enter the type of transport protocol used in the SDP rtpmap attribute. The default is RTP/AVP. Valid values are:

   - RTP/AVP | UDP

8. req-bandwidth—Enter the total bandwidth in kilobits that the media requires. The default value is 0. The valid range is:
   - Minimum—0
   - Maximum—4294967295
9. frames-per-packet—Enter the maximum number of frames to use per RTP packet. Leaving this parameters set to 0, its default value means that it is not being used. The valid range is:
   - Minimum—0
   - Maximum—256

   The interpretation of this value varies with codec type and with specific codec.
   - For frame-based codecs, the frame size is specific to each. For example, a G.729 frame contains ten milliseconds of audio, while a G.723.1 codec frame contains thirty milliseconds.
   - For sample-based codecs such as G.711, each frame contains one millisecond of audio.
10. parameters—Enter additional codec information. For example, the G.723.1 codec can have an additional silenceSuppression parameter.
11. average-rate-limit—Enter the maximum speed in bytes per second for the flow that this media profile applies to. The default value is 0. The valid range is:
    - Minimum—0
    - Maximum—125000000
12. peak-rate-limit—Enter the peak rate for RTP flows in bytes per seconds. The default is 0. The valid range is:
    - Minimum—0
    - Maximum—125000000
13. max-burst-size—Enter the maximum data size at peak rate in bytes. The default is 0. The valid range is:
    - Minimum—0
    - Maximum—125000000
14. sdp-bandwidth—Enable this parameter to use the AS bandwidth modifier in the SDP in the conditions for the application specific bandwidth modifier. The default is disabled. Valid values are:
    - enabled | disabled
15. sdp-rate-limit-headroom—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the average-rate-limit (rate limit for the RTP flow). The default is 0. The valid range is:
    - Minimum—0
    - Maximum—100

Fast Start/Slow Start Configurations

When you configure an H.323 interface, you configure it for either Fast Start to Slow Start translation or for Slow Start to Fast Start translation. You cannot configure one H.323 interface for both translation modes.

In the ACLI, you will set the following:

```
fs-in-first-msg       Fast Start must be sent in 1st response to Setup message
call-start-fast       Enable outgoing Fast Start call
call-start-slow       Enable outgoing Slow Start call
media-profiles        list of default media profiles used for outgoing call
```

To configure H.323 interfaces for Fast Start/Slow Start translations:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a ? at the system prompt. The display also includes H.323 service mode parameters.

5. fs-in-first-msg—Enable this parameter if you want to include Fast Start fields in the first message that the Oracle Enterprise Session Border Controller uses to respond to a Setup message. Usually, the first message sent is a Proceeding message If you do not want Fast Start fields included, leave this parameter set to its default value disabled. Valid values are:

   - enabled | disabled

6. call-start-fast—Enable this parameter if you want Slow Start calls to be translated to Fast Start when this H.323 interface is chosen as the outgoing interface. If this parameter is enabled, call-start-slow has to remain disabled. The default value is enabled. Valid values are:

   - enabled | disabled

     If you set this parameter set to disabled (default), the outgoing call will be set up in the same mode as the incoming call.

7. call-start-slow—Enable this parameter if you want Fast Start calls to be translated to Slow Start when this H.323 interface is chosen as the outgoing interface. If this parameter is enabled, call-start-fast has to remain disabled. The default value is disabled. Valid values are:

   - enabled | disabled

     If you leave this parameter set to disabled, the outgoing call will be set up in the same mode as the incoming call.

8. media-profiles—Enter the list of media profiles that you want to use when translating Slow Start calls to Fast Start. This information is used to open logical channels for the outgoing call.

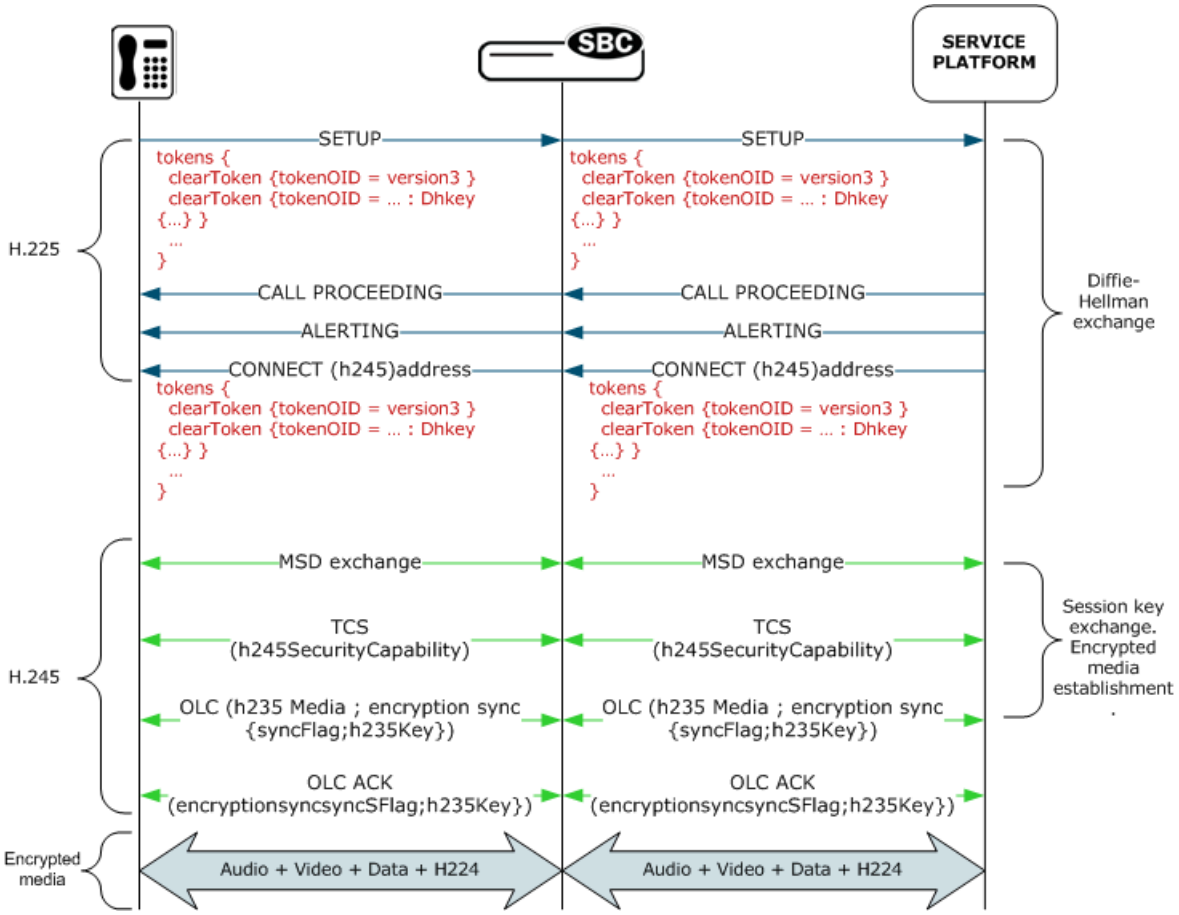
   If you enter the name of a media profile that does not exist, the Oracle Enterprise Session Border Controller will not perform translation. If you leave this parameter empty, the Oracle Enterprise Session Border Controller will not perform translation.

H.235 Encryption

Following the ITU-T H.235 encryption standard, the Oracle Enterprise Session Border Controller allows media (audio, video, and data) media that has already been encrypted by endpoints to pass through it, thereby supporting videoconferencing applications where media confidentiality is key. The ITU-T standard provides a profile with key management using Diffie-Hellman keys and the specification of an encryption algorithm.

Specifically, the Oracle Enterprise Session Border Controller permits the following:

- H.225 Setup and connect—The tokens parameter and its subfields in H.225 Setup and Connect message to pass transparently through the Oracle Enterprise Session Border Controller
- H.245Teminal CapabilitySet—The H.245 TerminalCapabilitySet messages to pass transparently through the Oracle Enterprise Session Border Controller, including:
  - Audio, video, and data capabilities
  - The h235SecurityCapability capability
- H.245 OpenLogicalChannel and OpenLogicalChannelAck—OLC messages with dataType h235Media to pass transparently through the Oracle Enterprise Session Border Controller; to accomplish this, the Oracle Enterprise Session Border Controller uses the mediaType subfield instead of the dataType field when the dataType is h235Media. The encryptionSync parameter and its subfields found in OLC and OLCAck messages to pass transparently through the Oracle Enterprise Session Border Controller.



You do not need to follow special configuration steps to enable this functionality; it works automatically.

RFC 2833 DTMF Interworking

This section explains the Oracle Enterprise Session Border Controller's support of transporting Dual Tone Multi-Frequency (DTMF) in Real-Time Transport Protocol (RTP) packets (as described in RFC 2833) to H.245 User Input Indication (UII).

Multimedia devices and applications must exchange user-input DTMF information end-to-end over IP networks. The Oracle Enterprise Session Border Controller provides the interworking capabilities required to interconnect networks that use different signaling protocols. Also, the Oracle Enterprise Session Border Controller provides DTMF translation to communicate DTMF across network boundaries.

The Oracle Enterprise Session Border Controller supports RFC 2833 to H.245 UII translation for H.323-to-H.323 calls, when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device that only uses H.245 UII.

About RFC 2833

RFC 2833 specifies a way of encoding DTMF signaling in RTP streams. It does not encode the audio of the tone itself, instead a signal indicates the tone is being sent. RFC 2833 defines how to carry DTMF events in RTP packets. It defines a payload format for carrying DTMF digits used when a gateway detects DTMF on the incoming messages and sends the RTP payload instead of regular audio packets.

About H.245 UII

H.245 provides a capability exchange functionality to allow the negotiation of capabilities and to identify a set of features common to both endpoints. The media and data flows are organized in logical channels. H.245 provides logical channel signaling to allow logical channel open/close and parameter exchange operations. The H.245 signaling protocol is reliable, which ensures that the DTMF tones will be delivered.

H.245 User Input Indication (UII) plays a key role in all the services that require user interaction. For video messaging, typical uses of UII include selection of user preferences, message recording and retrieval, and typical mailbox management functions. H.245 UII provides two levels of UII, alphanumeric and signal.

About 2833 to H.245 UII Interworking

The Oracle Enterprise Session Border Controller provides 2833 to H.245-UII interworking by checking 2833-enabled RTP streams for packets matching the payload type number for 2833. It then sends the captured packet to the host for processing and translation to H.245 UII messages. A H.245 UII message received by the Oracle Enterprise Session Border Controller is translated to 2833 packets and inserted into the appropriate RTP stream.

Flow Control Mapping for Interworking Function (IWF) Video

H.245 is a protocol for the transmission of call management and control signals in networks using H.323 equipment. The H.245 specification is used in audio, video, and data transmissions, as well as in voice over IP (VoIP). H.245 messages are sent over special channels called H.245 control channels.

H.245 signaling is used to manage and control call setup and connection. Functions of H.245 include determining which endpoint is to be the master and which is to be the slave during the call, opening and closing of multiplexed data-transfer paths between the endpoints, establishing an upper limit to the data transfer speed on each logical channel, information exchanges between endpoints concerning the types of data each endpoint can send and receive, requests by the receiving endpoint for changes in the mode of the data sent by the transmitting endpoint, and requests by either endpoint to end the call.

In the H.245 standard, the FlowControlCommand message is used to specify the upper limit of bit rate of either a single logical channel or the whole multiplex. The following is an excerpt from the H.245 standard.

Command Message: Flow Control (from H.245 standard)

```
==================================================
FlowControlCommand ::= SEQUENCE
{
        scope CHOICE
    {
    logicalChannelNumber LogicalChannelNumber,
    resourceID INTEGER (0..65535),
    wholeMultiplex NULL
    },
    restriction CHOICE
    {
    maximumBitRate INTEGER (0..16777215), -- units 100 bit/s
    noRestriction NULL
    },
    ...
}
==================================================
```

A terminal may send this command to restrict the bit rate that the far-end terminal sends. A receiving terminal must comply with this command.

In an H.323 environment, the Oracle Enterprise Session Border Controller previously used the FlowControlCommand to map to SIP using either the Real-Time Control Protocol (RTCP) feedback function, or the SIP signaling path (for example, the INFO method).

The Oracle Enterprise Session Border Controller now supports the SIP counter part of the H.245 FlowControlCommand using the SIP signaling path with the INFO method. The Oracle Enterprise Session Border Controller sends the SIP INFO message with "change_bitrate" rate parameter that has the value 100* maxBitRate from the corresponding H.245 FlowControlCommnad message. For example, in the following messages, the incoming H.323 message with the H.245 FlowControlCommand, is converted into the outgoing SIP INFO message with the message body.

Incoming H.323 Message with H.245 FlowControlCommand:

```
H.245
PDU Type: command (2)
        command: flowControlCommand (4)
            flowControlCommand
                scope: logicalChannelNumber (0)
                    logicalChannelNumber: 102
                restriction: maximumBitRate (0)
                    maximumBitRate: 4480
```

Outgoing SIP INFO Message:

```
Message Body
        eXtensible Markup Language
            <?xml
                version="1.0"
                encoding="utf-8"
                ?>
            <media_control>
                <vc_primitive>
                    <to_encoder>
                        <change_bitrate>
                            4480000
                            </change_bitrate>
                        </to_encoder>
                    </vc_primitive>
                </media_control>
```

About DTMF Transfer

DTMF transfer is the communication of DTMF across network boundaries. It is widely used in applications such as interactive voice response (IVR) and calling card applications.

The multiple ways to convey DTMF information for packet-based communications include:

- In-band audio: DTMF digit waveforms are encoded the same as voice packets. This method is unreliable for compressed codecs such as G.729 and G.723
- Out-of-band signaling events:

  H.245 defines out-of-band signaling events (UII) for transmitting DTMF information. The H.245 signal or H.245 alphanumeric methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of through the RTP channel. The tones are transported in H.245 UII messages.

  All H.323 version 2 compliant systems are required to support the H.245 alphanumeric method, while support of the H.245 signal method is optional.
- RTP named telephony events (NTE): uses NTE to relay DTMF tones, which provides a standardized means of transporting DTMF tones in RTP packets according to section 3 of RFC 2833.

Of the three RTP payload formats available, the Oracle Enterprise Session Border Controller supports RTP NTE.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for 2833 telephone event packets using a SDP or H.245 Terminal Capability Set (TCS), which tells the other side what payload type number to use when sending the named event packets to the local device. Most devices use payload type number 101 for 2833 packets, although no default is specified in the standard.

The 2833 packet's RTP header also makes use of the timestamp field. Because events often last longer than the 2833 packets sending interval, the timestamp of the first 2833 packet an event represents the beginning reference time for subsequent 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

See RFC 2833 and draft-ietf-avt-rfc2833bis-07.txt for more information.

Preferred and Transparent 2833

To support preferred (signaled) 2833 and transparent 2833, the Oracle Enterprise Session Border Controller provides 2833 detection and generation (if necessary) when the endpoint signals support for 2833.

- Preferred: the Oracle Enterprise Session Border Controller only generates and detects 2833 for endpoints if they negotiate support for 2833 through signaling
- Transparent: the Oracle Enterprise Session Border Controller behaves as it has prior to this release, offering and answering based on end-to-end signaling and transparently relaying 2833

Preferred 2883 Support

If one side of the call, or a session agent, is configured for preferred 2833, the Oracle Enterprise Session Border Controller only generates and detects 2833 for endpoints if they signal support for 2833. The Oracle Enterprise Session Border Controller will offer 2833 in the TCS SDP, even if the originating caller did not.

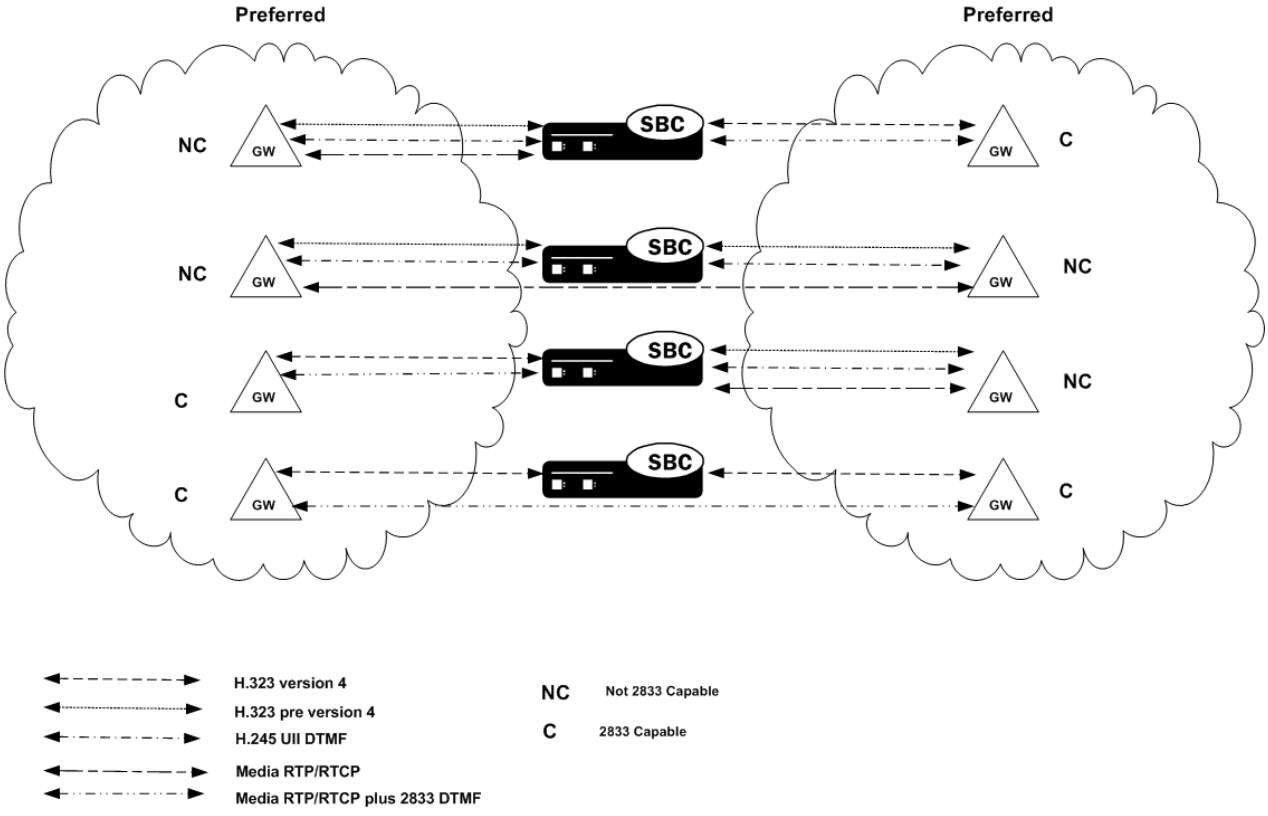
- When the Oracle Enterprise Session Border Controller manages calls originating from a preferred source going to a preferred target, it:

  Performs 2833 translation for an endpoint when the originating side requests 2833 but the target does not negotiate 2833

  Allows 2833 to pass through if the originating side and target of the call are configured as preferred and negotiate 2833
- When the Oracle Enterprise Session Border Controller manages calls originating from a preferred source going to a transparent target, it:

  Performs 2833 translation when the originating side requests 2833 but the target is configured as transparent and does not negotiate 2833.

  Allows 2833 to pass through if the originating side and the target of the call are configured as transparent and negotiate 2833. The Oracle Enterprise Session Border Controller does not perform active translation because both ends support 2833.

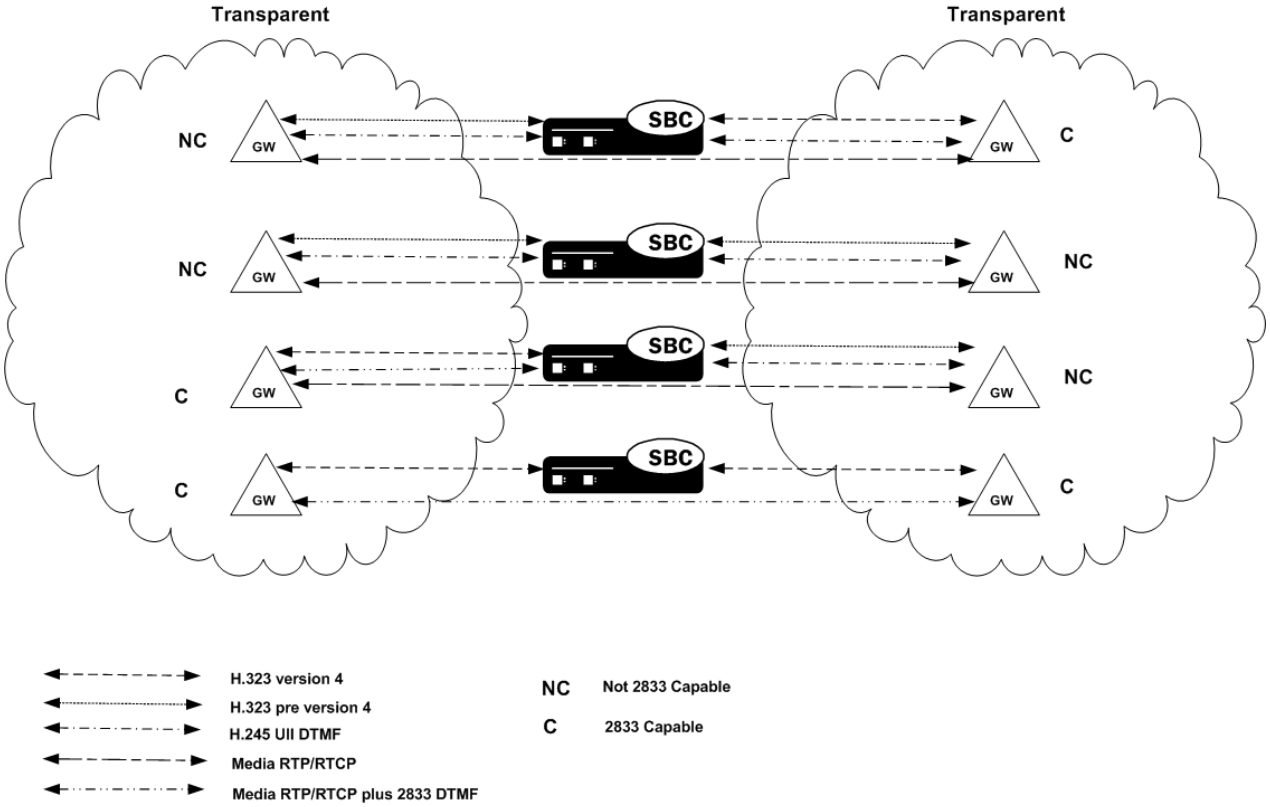


Transparent 2833 Support

The default configuration of the Oracle Enterprise Session Border Controller for H.323 is transparent 2833. The Oracle Enterprise Session Border Controller passes on the offered capabilities to the next-hop signaling element. If the next-hop endpoint is for a transparent 2833 target, typical capability negotiation determines the DTMF method. The Oracle Enterprise Session Border Controller transparently relays the DTMF as it has in previous releases.

With transparent 2833, the Oracle Enterprise Session Border Controller acts as a typical B2BUA or B2BGW/GK. However when the target of the call is configured as preferred 2833, the Oracle Enterprise Session Border Controller:

- Relays the 2833 packets if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target signals 2833
- Performs 2833 translation if the originating endpoint does not signal 2833 and the next-hop endpoint for the preferred target does signal 2833
- Does not perform 2833 translation or transparently relay 2833 if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target (or even a transparent 2833 target) does not signal 2833.



Basic RFC 2833 Negotiation Support

If H.323 or session agents on either side of the call are configured for preferred 2833 support, the Oracle Enterprise Session Border Controller supports end-to-end signaled negotiation of DTMF on a call-by-call basis. If the calling party is not configured for preferred support but sends 2833, the Oracle Enterprise Session Border Controller sends 2833 to the next-hop called party. If the calling party sends H.245 signals or alphanumeric UII, the Oracle Enterprise Session Border Controller sends H.245 signals or alphanumeric UII to the next-hop called party (if it is an H.323 next-hop).

The Oracle Enterprise Session Border Controller also supports hop-by-hop negotiation of DTMF capability on a call-by-call basis, if the signaling protocols or session agents on either side of the call are configured for preferred 2833 support.

H.323 to H.323 Negotiation

The Oracle Enterprise Session Border Controller serves as the H.323 called gateway. It answers RFC 2833 audio telephony event capability in the version 4 H.323/H.245 TCS when it receives a call from an H.323 endpoint configured for preferred RFC 2833.

If the Oracle Enterprise Session Border Controller is the answering device, configured for preferred support, and the calling device sends 2833, the Oracle Enterprise Session Border Controller accepts the 2833 regardless of the next-hop's DTMF capabilities. The received dynamic RTP payload type is used for detecting 2833 packets, while the response dynamic payload type is used for generating 2833 packets.

The Oracle Enterprise Session Border Controller supports:

- RFC-2833 audio telephony events in the version 4 H.323/H.245 TCS as the H.323 calling gateway, when the Oracle Enterprise Session Border Controller calls an H.323 endpoint configured for preferred RFC 2833 support. The Oracle Enterprise Session Border Controller sends 2833 to the called party regardless of whether the calling party sends it.
- H.245 UII and RFC-2833 packets sent at the same time, to the same endpoint, even if only half of the call is being provided 2833 support by the Oracle Enterprise Session Border Controller.

  If one half of the call supports H.245 UII, and the other half is being provided 2833 translation by the system, the Oracle Enterprise Session Border Controller can also forward the H.245 UII it receives to the 2833 endpoint. For example, when the signaling goes through a gatekeeper or third party call control, sending the H.245 UII in the signaling path allows those devices to learn the DTMF digits pressed.

Signal and Alpha Type Support

The Oracle Enterprise Session Border Controller supports:

- H.245 signal and alpha type UII in the H.323/H.245 TCS as the H.323 calling gateway when the:

  Oracle Enterprise Session Border Controller calls an H.323 endpoint configured for transparent 2833 support

  calling endpoint's target is configured as preferred

  If the originating preferred side also sends 2833, the Oracle Enterprise Session Border Controller forwards it to the transparent side. The Oracle Enterprise Session Border Controller sends signal and alpha UII support to the called party regardless of whether the calling party sends it, if the call originates from a preferred side to a transparent side.
- H.245 alphanumeric UII for DTMF for H.323 endpoints that do not signal 2833 or contain explicit H.245 UII capability, for stacks configured for transparent 2833 support.

  When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Oracle Enterprise Session Border Controller translates incoming H.245 UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Oracle Enterprise Session Border Controller relays the H.245 UII messages.
- H.245 signal type UII for DTMF for H.323 endpoints that do not signal 2833, but do signal explicit H.245 UII capability, for stacks configured for transparent 2833 support.

  When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Oracle Enterprise Session Border Controller translates incoming H.245 signaled UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Oracle Enterprise Session Border Controller relays the H.245 UII messages if both sides support it.

H.323 Endpoints

Because there are different H.323 endpoints based on different versions of H.323, the DTMF can be either be transferred out-of-band as UII or in-band using RFC 2833. Most H.323 endpoints:

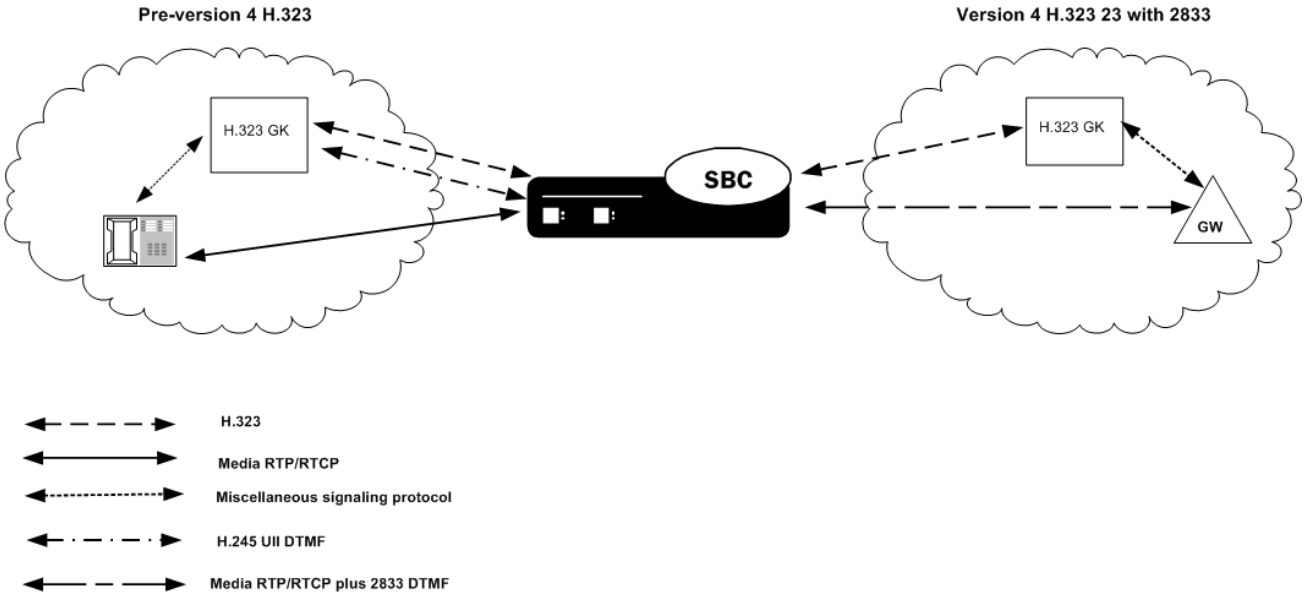
- version 4 and above support RFC 2833
- version 2 and pre-version 4 support UII-Signal
- version 1 and pre-version 2 support UII-Alphanumeric

Translating H.245 UII to 2833 for H.323 Calls

A majority of H.323 endpoints are not version 4 H.323 compliant and do not support RFC 2833 for DTMF transfer. However, some networks include version 4 H.323 devices that require the DTMF events to be signaled in 2833 packets. Network-based version 4 H.323 gateways use RFC 2833 instead of H.245 UII. (Version 4 H.323 devices should support H.245 UII.)

The Oracle Enterprise Session Border Controller translates 2833 to H.245 UII for H.323-to-H.323 calls when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device which only uses H.245 UII.

The Oracle Enterprise Session Border Controller can translate H.245 UII to RFC2833 and back, based on the admin configuration and H.245 TCS exchanges. This translation enables DTMF to work end-to-end.



RFC 2833 Mode Configuration

To configure RFC 2833 mode:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type h323-stacks and press Enter.

   ```
   ACMEPACKET(h323)# h323-stacks
   ACMEPACKET(h323-stack)#
   ```

   From this point, you can configure H.323 stack parameters. To view all H.323 stack parameters, enter a ? at the system prompt.

5. rfc2833-mode—Set the RFC2833 mode. The default value is transparent. The valid values are:

   - transparent:—The Oracle Enterprise Session Border Controller and H.323 stack behave exactly the same way as before and the 2833 or UII negotiation is transparent to the Oracle Enterprise Session Border Controller.
   - preferred—The H323 stack uses 2833 for DTMF transfer, which it signals in its TCS. However, the remote H323 endpoint makes the decision. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack reverts back to using UII. You configure the payload format by configuring the h323-config element.

RFC 2833 Payload Configuration

To configure the RFC 2833 payload in preferred mode:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# h323
   ```

   From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt.

4. rfc2833-payload—Enter a number that indicates the payload type the Oracle Enterprise Session Border Controller will use for RFC 2833 packets while interworking 2833 and UII. The default value is 101.The valid range is:

   - Minimum—96
   - Maximum—127

   You configure session agents with:

   - payload type the Oracle Enterprise Session Border Controller wants to use for RFC 2833 packets while interworking 2833 and UII.

     The default value for this attribute is 0. When this value is zero, the global rfc2833-payload configured in the h323-configuration element will be used instead. For SIP session agents, the payload defined in the SIP interface is used, if the SIP interface is configured with the preferred RFC 2833 mode.
   - 2833 mode

     A value of transparent or preferred for the session agent's 2833 mode will override any configuration in the h323-stack configuration element.

RFC 2833 SA Configuration

To configure session agents:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. rfc2833-mode—Set the RFC 2833 mode you want the session agent to use. The default is none. The valid values are:

   - none—2833 to UII interworking is based on the H.323 stack configuration.
   - transparent:—The 2833 or UII negotiation is transparent to the Oracle Enterprise Session Border Controller. This overrides the H.323 stack configuration, even if the stack is configured for preferred mode.
   - preferred:—2833 for DTMF transfer is preferred, which is signaled in the TCS. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack configured as preferred will revert back to using UII. This overrides any configuration in the h323-stack even if the stack is configured for transparent mode.

5. rfc2833-payload—Enter a number that indicates the payload type the session agent will use for RFC 2833 packets while interworking 2833 and UII. The default value is 0. The valid range is:

   - Minimum—0, 96

- Maximum—127

H.323 Registration Proxy

The Oracle Enterprise Session Border Controller provides a registration proxy feature that allows a gatekeeper to authenticate a registration before accepting it. This feature is key when two factors are present: authentication is required, and an RRQ from an endpoint includes a token and/or cryptographic token. If authentication for that endpoint is to work, the Oracle Enterprise Session Border Controller must forward the registration requests received from the endpoint to the gatekeeper separately.When you do not use the H.323 registration proxy, the Oracle Enterprise Session Border Controller combines all registrations received from H.323 endpoints into a single RRQ and sends it to the gatekeeper. Using the H.323 registration proxy, you can configure the Oracle Enterprise Session Border Controller to use separate forwarding.

When registration requests are forwarded separately, each RRQ must have a unique CSA. This means that the Oracle Enterprise Session Border Controller must perform a one-to-one translation of the CSA in the incoming RRQ to a distinct transport address. The translated address replaces the endpoint's CSA in the outgoing RRQ. Then the Oracle Enterprise Session Border Controller must listen for incoming calls that arrive at this translated transport address for the registered endpoint.

H.235 Authentication Transparency

When operating in this mode, H.235 authentication tokens (cryptotokens) in RAS messages proxied though the Oracle Enterprise Session Border Controller are passed through transparently.

For applications where Oracle Enterprise Session Border Controller is between H.323 gateways and a network hosted gatekeeper, the H.235 cryptotokens are passed through unmodified in RAS messages: RRQs, ARQs, and DRQs. This feature allows for secure gateway authentication.

Unique CSA Per Registered Gateway

When operating in this mode, each CSA is mapped to a registered gateway for call routing. The core gatekeeper does not support additive registrations, so a different CSA must be used for each unique registration that goes to the gatekeeper. The gatekeeper does not overwrite previously registered aliases. Also, since the gatekeeper initiates calls to an endpoint on the CSA specified in the RRQ, the Oracle Enterprise Session Border Controller must listen on the assigned address for incoming calls to that client as long as the client is registered.

Virtual Call Signaling Address

You can configure the Oracle Enterprise Session Border Controller with:

- A TCP port range for Q.931—Q.931 ports that are frontend ports handled by a real backend socket, and are therefore virtual
- ATCP port range for separate H.245 TCP connections—Actual sockets that the Oracle Enterprise Session Border Controller handles separately

Virtual call signaling address is an H.323 call signaling address that is registered with a gatekeeper, but does not have a corresponding listening socket in the Oracle Enterprise Session Border Controller. Using the virtual call signaling address means that numerous network transport addresses do not need to be allocated.

Virtual call signaling addresses work by attaching a range of TCP server ports to a single listening TCP socket. After a connection is accepted, the accepting socket created by the server socket operated normally, as though it were created by the server socket that listens on the same transport address as the destination of the arriving packet.

To use virtual call signaling addresses, you specify a Q.931 port range from which the Oracle Enterprise Session Border Controller can allocate ports. This port range is associated with the virtual call signal IPv4 address you specify. To bind dynamic TCP connections to aport within a port range, you configure a dynamic H.245 port range. The dynamic H.245 port range refers to the separate TCP connection for H.245 that takes place when tunneling is not being used. This enables the Oracle Enterprise Session Border Controller to select the port to which the TCP socket is bound. These two port ranges cannot overlap.

When a new RRQ has to be forwarded to the gatekeeper, the Oracle Enterprise Session Border Controller caches the registration and then forwards a modified copy of the RRQ. The Oracle Enterprise Session Border Controller

allocates a virtual call signal address on the gateway stack and uses it to replace the CSA of the registering endpoint in the forwarded RRQ.

Virtual RAS Address

The Oracle Enterprise Session Border Controller also allocates a virtual RAS address for each endpoint registration. Before forwarding an RRQ from an endpoint, the Oracle Enterprise Session Border Controller replaces the RAS address of the registering endpoint with the virtual RAS address on the gateway interface.

RAS Message Proxy

When the Oracle Enterprise Session Border Controller's registration proxy feature is configured, RAS messages to and from endpoints are forwarded, except for the following: GRQ, GCF, GRJ, IRQ, IRR, IACK, and INACK. If the system receives a valid GRQ on the RAS port of the gatekeeper stack that supports H.323 registration, it responds with a GCF message. Otherwise, it sends a GRJ message.

If the gateway interface receives IRR or IRQ messages, the Oracle Enterprise Session Border Controller attempts to respond based on the information about the call, and does not forward the messages.

Other RAS messages are forwarded after some modifications:

- Translating the transport address
- Deleting fields that the Oracle Enterprise Session Border Controller does not support

About Setting Port Ranges

When you configure the H.323 registration proxy feature, you set the Q.931 port range and the dynamic H.245 port range for H.245 connections. If you configure a Q.931 port range, you must also configure a dynamic H.245 port range.

These port ranges cannot overlap because of TCP ports must be unique. The dynamic H.245 port range is used to allocate a real TCP socket, but the Q.931 port range allocates a virtual call signaling address that does not have an associated listening TCP socket.

Note: You should choose these sockets with future Oracle Enterprise Session Border Controller features about security in mind because future development will support performing admission control based on these port ranges. You will be able to set up filtering rules to allow only inbound packets to configured port ranges.

The following table shows how the Q.931 and dynamic H.245 port ranges work. If you set the start port of 1024 and the number of ports to 1024, you will have configured a port range that starts at 1024 and ends at 2047. So the final port in the range is the start port number added to the number of points, minus 1. Remember that you cannot overlap the Q.931 and dynamic H.245 port ranges. Notice that the higher the number of the start ports, the fewer ranges of ports you have remaining from which to choose.

Number of Ports	Start Port	n
1024	1024 * n	1-63
2048	2048 * n	1-31
4096	4096 * n	1-15
8192	8192 * n	1-7
16384	16384 * n	1-3
32768	32768 * n	1

H.323 Registration Proxy Configuration

In the ACLI, the parameters that apply to this feature are:

```
q931-start-port       Starting port number for port range used for Q.931
call signalling
q931-number-ports     Number of ports in port range used for Q.931 call
```

```
signalling
dynamic-start-port    Starting port number for port range used for
dynamic TCP connections
dynamic-number-ports  Number of ports in port range used for dynamic TCP
connections
```

To configure the H.323 registration proxy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type h323-stack and press Enter.

   ```
   ACMEPACKET(h323)# h323-stacks
   ACMEPACKET(h323-stack)#
   ```

5. q931-start-port—Enter the number where you want the Q.931 port range to start. The default value is 0. Valid values are:

   - 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

6. q931-number-ports—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. The default value is 0. Valid values are:

   - 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

   Note: If you have enabled process registration for this H.323 interface, this value must be set to zero because the interface is a gatekeeper that does not support the virtual call signaling address feature.

7. dynamic-start-port—Enter the number where you want the dynamic H.245 port range to start. The default value is 0. Valid values are:

   - 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

8. dynamic-number-ports—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. The default value is 0. Valid values are:

   - 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

H.323 Registration Caching

The Oracle Enterprise Session Border Controller can cache and proxy an H.225 RRQ between an H.323 endpoint and a gatekeeper. Registration caching has two benefits:

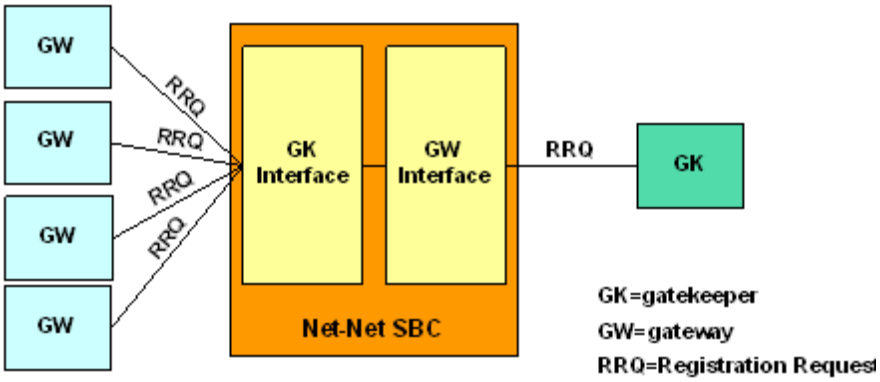
- It allows the aggregation of RRQs sent to a gatekeeper stack and proxies those requests through the gateway stack. If the external gatekeeper associated with the outbound (gateway) interface does not support additive registration, then the Oracle Enterprise Session Border Controller consolidates the requests by placing them all in the same packet. Otherwise, additive registration is used on the outbound (gateway) interface.
- It allows the gatekeeper stack to use the registration information to route calls from other realms to the endpoints in its realm.

For registration caching, you need to configure at least two H.323 interfaces:

- One gatekeeper interface to receive registrations
- One gateway interface to proxy registrations

The Oracle Enterprise Session Border Controller caches all successful registrations, using the cache to route calls back to the associated endpoint.

The following diagram shows how RRQs flow during registration caching.



Caveats for Registration Caching

This feature has the following caveats:

- If a gateway stack receives a URQ message from the gatekeeper, it confirms the request with an UCF message. It flushes all registration caching for that stack. However, the Oracle Enterprise Session Border Controller does not send URQs to the registered endpoints.
- The Oracle Enterprise Session Border Controller must be rebooted so that the gateway interface can rediscover the gatekeeper under the following circumstances:

Automatic gateway discovery is turned on for the gateway interface by setting the automatic gateway discovery parameter to enabled.

Configuration Requirements

For the Oracle Enterprise Session Border Controller to determine where to route an RRQ, either the associated stack parameter or the gatekeeper identifier field is used.

First, the Oracle Enterprise Session Border Controller uses the associated interface (assoc-stack) of the gatekeeper interface to find the interface for the outgoing RRQ. If you do not configure an associated interface and the incoming RRQ has a gatekeeperIdentifier field, the Oracle Enterprise Session Border Controller finds a configured gateway interface with a matching gk-identifier field and use it as the outgoing interface. If the incoming RRQ does not have a gatekeeperIdentifier field and the gatekeeper interface has a configured gatekeeper identifier, the Oracle Enterprise Session Border Controller finds a gateway interface with a gatekeeper identifier that matches the one set for the gatekeeper interface and then use it as the outgoing interface. If an outgoing interface cannot be determined, the Oracle Enterprise Session Border Controller rejects the RRQ with the reason discoveryRequired.

A configured H.323 interface can be the gateway interface for more than one gatekeeper interface. If a call is received on the gateway interface, the registration cache will be queried to find a registration matching the call's destination. If a registration is found, the interface on which the registration was received will be used as the outgoing interface for the call.

Subsequent ARQ or URQ messages coming from a registered endpoint will be proxied to the gatekeeper using the outgoing gateway interface established during registration. If a registration is not found, an ARJ or a URJ will be sent to the endpoint originating the ARQ or URQ.

A gatekeeper interface can respond to a GRQ if the GRQ is received on its RAS interface. The Oracle Enterprise Session Border Controller supports GRQ on a multicast address.

H.323 Registration Caching Configuration

In the ACLI, the parameters that apply to this feature are:

```
isgateway               Enable the stack to run as a gateway
registration-ttl        Number of seconds before the registration becomes
invalid
terminal-alias          List of aliases for terminal
gatekeeper              Gatekeeper's address and port
gk-identifier           Gatekeeper's identifier
```

To configure the gateway interface parameters for registration caching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stack and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

5. isgateway—Enable H.323 stack functionality as a Gateway. Leave this parameter set to its default, enabled, so the H.323 stack runs as a Gateway. When this field is set to disabled, the H.323 stack runs as a Gatekeeper proxy.Leave this parameter for the service mode set to it's default, enabled. Valid values are:

   - enabled | disabled

     Enabling this parameter ensures that registration with the gatekeeper upon startup. It also ensures that all calls will be preceded by an ARQ to the gatekeeper for admission control.

6. registration-ttl—Set the registration expiration parameter to the value of the timeToLive field in the RRQ sent to the gatekeeper. The default is 120. The valid range is:

   - Minimum—0
   - maximum—4294967295

     When the Oracle Enterprise Session Border Controller receives an RCF from the gatekeeper, it extracts the timeToLive field and uses that value as the time interval for keeping the registration of the gateway interface alive. The Oracle Enterprise Session Border Controller sends a keep-alive RRQ about ten seconds before the registration expires.

     The registration expiration you set value should not be too low because some gatekeepers simply accept the timeToLive in the RRQ, resulting in a potentially high volume of RRQs.

7. terminal-alias—Set this parameter if the gatekeeper requires at least one terminal alias in an RRQ. On startup, the gateway interface registers with the gatekeeper using this terminal alias.

   When the Oracle Enterprise Session Border Controller forwards an RRQ from an endpoint and if the gatekeeper does not support additive registration, the RRQ has the interface's terminal alias, the aliases of the registering endpoint, and other aliases of all registered endpoints. Otherwise, the RRQ only contains the aliases of the registering endpoint.

8. gatekeeper and gk-identifier—Configure these parameters if you do not want the Oracle Enterprise Session Border Controller to perform automatic gatekeeper discovery. If the gatekeeper identifier is empty, then the Oracle Enterprise Session Border Controller learns the gatekeeper identifier from the gatekeeperIdentifier field in the GCF.

Configuring the Gatekeeper Interface for Registration Caching

In the ACLI, the parameters that apply to this feature are:

```
isgateway             Enable the stack to run as a gateway
gatekeeper            Gatekeeper's address and port
gk-identifier         Gatekeeper's identifier
registration-ttl      Number of seconds before the registration becomes
invalid
```

To configure the gatekeeper interface parameters for registration caching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stack and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

5. isgateway—Set this parameter to disabled to run the H.323 stack as a Gatekeeper proxy.
6. gatekeeper—Leave this parameter empty.
7. auto-discovery—Disable the Automatic Gatekeeper discovery feature upon start-up. Set this parameter to disabled.
8. gk-identifier—Set this parameter to the identification of the gatekeeper to which RRQs received on this interface must be proxied.
9. registration-ttl—Enter the number of seconds to set the timeToLive field in the RCF destined for an endpoint. If you do not configure another value, this timer is set to 120 seconds (default).

   This value should not be set too high or too low:

   - Setting a value that is too high causes the registration to be alive too long. If an endpoint reboots during this interval and re-registers with the same terminal aliases (but changes its call signaling address), the registration will be rejected with the reason duplicateAlias.
   - Setting a value that is too low puts an unnecessary load on the Oracle Enterprise Session Border Controller because it has to handle keep-alive registrations from the endpoint constantly, especially when there are many registered endpoints. If an endpoint does not set the timeToLive field in its RRQ, the registration of that endpoint will not expire.

     If an endpoint registers again without first unregistering itself (e.g., when it crashes and reboots), the Oracle Enterprise Session Border Controller rejects the registration using the reason duplicateAlias. The Oracle Enterprise Session Border Controller uses this reason when the endpoint's call signaling address (IP address and port) is changed but its terminal aliases remain the same.

ACLI Registration Caching Configuration Example

In the following example, the H.323 gatekeeper interface (h323-stack) is private and the gateway interface (h323-stack) is public.

```
h323-config
state                         enabled
log-level                     DEBUG
response-tmo                  4
connect-tmo                   32
h323-stack
name                          private
state                         disabled
realm-id                      private
assoc-stack                   public
local-ip                      192.168.200.99
max-calls                     200
max-channels                  4
registration-ttl              120
terminal-alias
prefixes
ras-port                      1719
auto-gk-discovery             disabled
multicast                     0.0.0.0:0
gatekeeper                    0.0.0.0:0
gk-identifier
q931-port                     1720
```

```
alternate-transport
q931-max-calls                  200
h245-tunneling                  disabled
fs-in-first-msg                 disabled
call-start-fast                 disabled
call-start-slow                 disabled
media-profiles
process-registration            enabled
anonymous-connection            disabled
proxy-mode
filename
h323-stack
name                            public
state                           enabled
isgateway                       enabled
realm-id                        public
assoc-stack                     private
local-ip                        192.168.1.99
max-calls                       200
max-channels                    2
registration-ttl                120
terminal-alias
prefixes
ras-port                        1719
auto-gk-discovery               disabled
multicast                       0.0.0.0:0
gatekeeper                      192.168.1.50:1719
gk-identifier                   gk-public.acme.com
q931-port                       1720
alternate-transport
q931-max-calls                  200
h245-tunneling                  disabled
fs-in-first-msg                 disabled
call-start-fast                 disabled
call-start-slow                 disabled
media-profiles
process-registration            disabled
anonymous-connection            disabled
proxy-mode
filename
```

H.245 Stage

The Oracle Enterprise Session Border Controller allows you to set the earliest stage in an H.323 call when the Oracle Enterprise Session Border Controller initiates the procedure to establish an H.245 channel for the call. If you have enabled H.245 tunneling by setting the h245-tunneling parameter to enabled, then you do not need to configure your system for this feature.

The Oracle Enterprise Session Border Controller initiates the H.245 procedure by either:

- Sending its H.245 address, or
- Creating a TCP connection to an H.245 address that it has received

You can set this parameter to any of the following stages of an H.323 call: setup, proceeding, alerting, connect, early, facility, noh245, and dynamic. With the exception of early, noh245, and dynamic, these values correspond to types of H.225/Q.931 messages. The dynamic value is described in detail in the next section.

When you configure the early value, your Oracle Enterprise Session Border Controller begins the H.245 procedure at the time the Setup message is sent or received, or when the Connect message is received.

While these values allows for some flexibility about when the H.245 process is started, they are inherently static. All calls in the H.323 stack configuration use the same value, and it cannot be changed from call to call on that stack.

Dynamic H.245 Stage Support

You can configure your Oracle Enterprise Session Border Controller for dynamic H.245 support, meaning that the point at which the H.245 process begins can be determined dynamically. To support dynamic H.245, the Oracle Enterprise Session Border Controller sends its H.245 address in the incoming call when it receives an H.245 address in the outgoing call.

Dynamic H.245 Stage for Incoming Calls

When a call comes in on an H.323 interface that you have configured for dynamic H.245 stage support.

The Oracle Enterprise Session Border Controller includes its H.245 address in the h245Address field of the first H.225/Q.931 message. The Oracle Enterprise Session Border Controller does this after it receives the first H.225/Q.931 message with an H.245 address in the outgoing call. Based on the first H.225/Q.931 message received by the Oracle Enterprise Session Border Controller that has an H.245 address, the Oracle Enterprise Session Border Controller selects the message in which to include the H.245 address as outlined in the table below.

Message Received with H.245 Address	Message Sent with H.245 Address
Call Proceeding	Call Proceeding, Progress, Alerting, Connect or Facility. The H.245 address is sent in the Call Proceeding message if the system has not sent a Call Proceeding message in the incoming call. This is true only when you enable the Fast Start in first message parameter for the incoming stack; this parameter establishes whether or not Fast Start information must be sent in the first response to a Setup message. Otherwise, the message in which the H.245 address is sent depends on what message is received after the Call Proceeding message. This is because the Oracle Enterprise Session Border Controller sends its Call Proceeding message directly after receiving the Setup message.
Progress	Progress
Alerting	Alerting
Connect	Connect
Facility	Facility

When it receives the first H.225/Q.931 message with an H.245 address in the outgoing call, the Oracle Enterprise Session Border Controller creates a listening socket on the incoming interface. It also includes the socket address and port in the H.245 address of the next H.225/Q.931 message that it sends. If there is no pending H.225/Q.931 message for the Oracle Enterprise Session Border Controller to send, it instead sends a Facility message with the reason startH245. Then the H.245 channel is established when a TCP connection is made to the listening socket.

For the outgoing leg of a call that came in on the H.323 stack configured for H.245 dynamic stage support, the Oracle Enterprise Session Border Controller starts establishing the H.245 channel when it receives the first H.225/Q.931 message with H.245 address information. It also starts to establish a TCP connection to the address and port specified in the H.245 address information. The H.245 channel for the outgoing call is established while the H.245 address (h245Address) is sent in the incoming call as described above.

Dynamic H.245 Stage for Outgoing Calls

This section describes what happens when a message exits the Oracle Enterprise Session Border Controller on an H.323 stack that you have configured for dynamic H.245 stage support.

When the Oracle Enterprise Session Border Controller receives the first H.225/Q.931 message that has H.245 address information, it establishes an H.245 channel. The Oracle Enterprise Session Border Controller initiates a TCP connection to the address and port specified in the H.245 address information.

If the incoming call for the session is also on an H.323 stack with dynamic H.245 configured, the Oracle Enterprise Session Border Controller starts the H.245 procedure in the incoming call. Otherwise, the system sends its H.245 address in the incoming call based on the H.245 stage support that you have configured.

The process is different when the Oracle Enterprise Session Border Controller receives a TCS message on the outgoing call before the incoming call reaches its H.245 stage. In this instance, the Oracle Enterprise Session Border Controller sends a Facility message with the reason startH245 with its H.245 address in order to start the H.245 procedure. The reason is needed in order for the Oracle Enterprise Session Border Controller to exchange TCS messages with the incoming side of the call.

H.245 Stage Configuration

To configure H.245 stage support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

5. h245-stage—Set this field to the stage at which the Oracle Enterprise Session Border Controller transfers the H.245 address to the remote side of the call, or acts on the H.245 address sent by the remote side. The default value is Connect. Valid values are:

   - Setup | Alerting | Connect | Proceeding | Early | Facility | noh245 | Dynamic

H.323 HNT

This section explains how H.323 hosted NAT traversal (HNT) works and how to enable this capability on your Oracle Enterprise Session Border Controller.

The feature enables endpoints behind NATs to originate and terminate calls by resolving the address differences between the NAT and the actual endpoint.

H.323 communication through a NAT becomes an issue when engaging in RAS messaging. While the H.323 standard specifies specific information elements in the RAS messages that indicate the address to which the replies should be sent, these addresses will be behind the NAT and therefore unroutable. The Oracle Enterprise Session Border Controller solves this problem by sending RAS replies to the layer 3 address from which the associated RAS request was received.

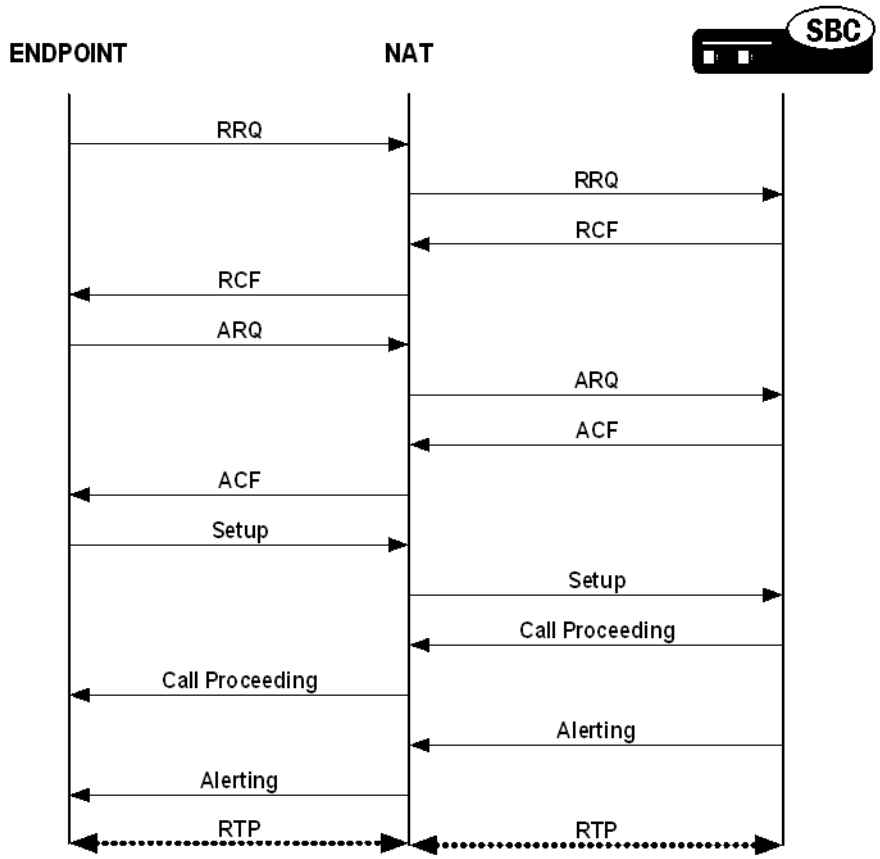
A second issue exists for media channels as the address specified in the H.323 OLC message will be behind the NAT and likewise unroutable. This is resolved by relying on the fact that the forward and reverse channels will utilize the same address and port on the endpoint. By sending media packets to the same address from which the packet are received, media and flow through the NAT.

If you do not use H.323 HNT, the following behavior will occur:

- When an H.323 endpoint is behind a NAT and it registers with a gatekeeper through the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller tries to send a response back to the endpoint's RAS address rather than to the NAT from which the request was received.
- The same is true for LRQ and IRQ messages because responses without H.323 HNT for outbound sessions, responses were being sent back to the replyAddress or the rasAddress.
- In addition, the Oracle Enterprise Session Border Controller always induces one-way media because it tries to send the RTP to the media IP address and port it receives in the OLC messages rather than the ephemeral port on the intermediary NAT.

With this ability enabled, however, the Oracle Enterprise Session Border Controller sends RAS responses back to the address from which the request was received (the NAT). It does not send responses to the endpoint's rasAddress or replyAddress mentioned in the signaling message. The same is true for RTP. With H.323 HNT for outbound sessions enabled, the Oracle Enterprise Session Border Controller sends RTP to the IP address and port from which it receives the RTP packets (the NAT).

The call flow below illustrates how this feature works:



Caveats

Keep in mind the following caveats when you are enabling H.323 HNT for outbound sessions on your Oracle Enterprise Session Border Controller:

- This capability does not apply to calls that require IWF translation between SIP and H.323.

H.323 HNT Configuration

You can enable this capability for specific H.323 interfaces.

To enable H.323 HNT:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type h323-stack and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters for the H.323 interface.

```
ACMEPACKET(h323)# h323-stack
```

5. If you are adding this service to a new H.323 interface that you are creating, type options hnt (to enable H.323 HNT), and then press Enter.

```
ACMEPACKET(h323-stack)# options hnt
```

6. If you are adding this service to an H.323 interface that already exists, type select to select the interface to which you want to add the service. Then use the options command and prepend the option with a plus (+) sign.
   - If you know the same of the interface, you can type the name of the interface at the name: prompt and press Enter.
   - If you do not know the name of the interface, press Enter at the name: prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press Enter.
   - If are adding service to an existing interface and you type options hnt without a plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

H.323 Party Number-E.164 Support

Some H.323 gateways cannot handle partyNumber alias addresses in H.225 messages. The system lets you convert this address type to dialedDigits (E.164). This conversion applies to sourceAddress, destinationAddress, and destExtraCallInfo aliases in Setup messages.

To enable this feature, use the convertPNToE164 value in the options field of the H.323 stack configuration.

Signaling Only Operation

When you set the Oracle Enterprise Session Border Controller to operate in signaling-only mode, it acts like a signaling server. It proxies the call signaling messages between two endpoints. Note, however, that the NOracle Enterprise Session Border Controller does not function as a RAS proxy; it does not proxy RAS messages.

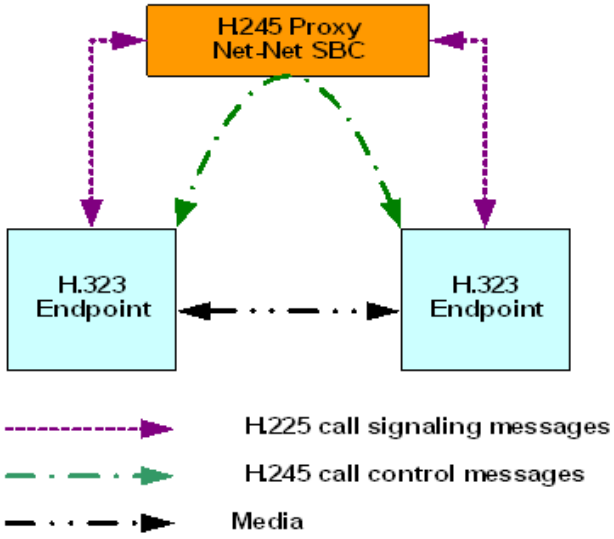
You have two options for the proxy mode:

- H.245 proxy mode—The Oracle Enterprise Session Border Controller handles call signaling (H.225) and call control (H.245) messages.
- H.225 proxy mode—The Oracle Enterprise Session Border Controller handles call signaling

To use this feature, you need to set the proxy mode parameter in the H.323 interface configuration to H.225 or H.245.

H.245

When in H.245 proxy mode, the Oracle Enterprise Session Border Controller proxies or passes through the call signaling (H.225) messages and the call control (H.245) messages. It allows media to flow between the two H.323 endpoints, as shown in the following diagram.



In some deployments, the media might be treated by a NAT device. When the Oracle Enterprise Session Border Controller is in H.245 proxy mode, any tunneled H.245 message on the ingress side is tunneled in the egress side. However, if the tunneling is refused on the egress side, a separate H.245 session is established.

H.245 proxy mode support is defined in the following table.

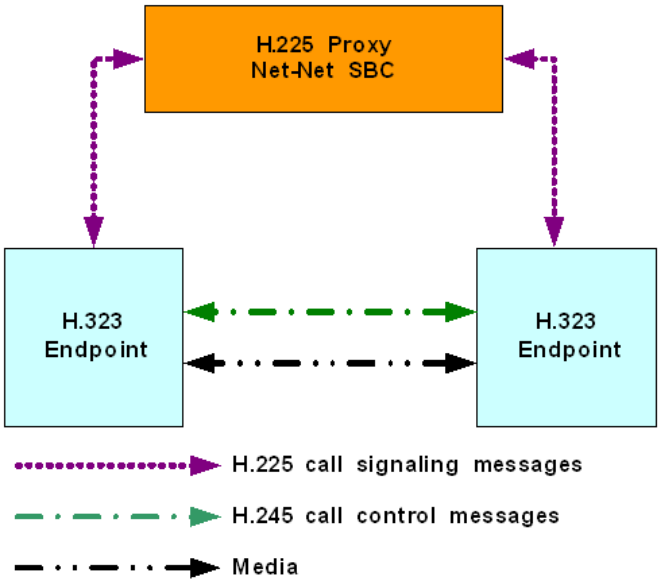
Ingress	Egress
Tunneled	Tunneled
Tunneled	Separate H.245 session
Separate H.245 session	Tunneled
Separate H.245 session	Separate H.245 session

H.225

When in H.225 proxy mode, the Oracle Enterprise Session Border Controller only proxies call signaling (H.225 messages). The call control (H.245 messages) and the media associated with the session do not go through the Oracle Enterprise Session Border Controller. Instead, they flow directly between the two H.323 endpoints.

Note: H.225 proxy mode is only used in specific applications and should not be enabled without consultation from your Acme Packet Systems Engineer.

The following diagram shows the flow.



In certain deployments, the call control message and media are exchanged between the two H.323 endpoints themselves. When the Oracle Enterprise Session Border Controller is in H.225 proxy mode, any tunneled H.245 message on the ingress side is tunneled in the egress side; this is irrespective of the value configured in the value you set for the h.245-tunneling parameter in the H.323 stack configuration.

Maintenance Proxy Function

The Oracle Enterprise Session Border Controller supports a maintenance proxy function for H.323 and enhances the way the Oracle Enterprise Session Border Controller creates unique RAS ports. You can register endpoints through the Oracle Enterprise Session Border Controller with unique RAS port. You can also set the H.323 interface on the enterprise side to represent enterprise-side endpoints and thereby register on the carrier side.

The maintenance proxy creates a many-to-one association between the enterprise and the carrier side. Interfaces on the enterprise side can be associated with the carrier side interface, which also must be configured to for the maintenance proxy feature.

You configure the maintenance proxy feature by simply setting an option in the H.323 interface configuration.

Maintenance Proxy Configuration

To configure the maintenance proxy function, you need to set two values in the options parameters for the H.323 interface (h323-stack):

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

5. options—Set the options parameter to maintenanceProxy.

Applying TCP Keepalive to the H.323 Interface

To apply these settings individually per H.323 interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stack and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)# tcp-keepalive
```

5. tcp-keepalive—Disable this parameter if you do not want the TCP keepalive network parameters to be applied. The default value is disabled. Valid values are:

   - enabled | disabled

6. Click OK at the bottom of the window to complete configuring TCP keepalives and the maintenance proxy.

Automatic Gatekeeper Discovery

Available only when the H.323 interface is functioning as a gateway, this feature allows for automatic gatekeeper discovery on start-up.

This feature is based on the Oracle Enterprise Session Border Controller sending a GRQ to the multicast address of the RAS Multicast Group, which is the device group listening on this address. If you do not configure a multicast address, Oracle Enterprise Session Border Controller uses the well-known address and port 224.0.1.41:1718 in the address-port combination making up this parameter.

Multicast only functions when the Oracle Enterprise Session Border Controller is discovering an external gatekeeper. The Oracle Enterprise Session Border Controller does not respond to multicast gatekeeper queries.

When it receives a GCF message from a gatekeeper, the Oracle Enterprise Session Border Controller registers with the gatekeeper indicated in the GCF. When it receives an GRJ message that contains optional information about alternative gatekeepers, the Oracle Enterprise Session Border Controller attempts to register with an alternate.

If you do not use automatic gatekeeper discovery, the Oracle Enterprise Session Border Controller registers with the gatekeeper you configure in the gatekeeper parameter. In this case, the gatekeeper identifier you configure is included in to the RRQ. No registration a takes place if you do not establish automatic gatekeeper discovery or if you do not configure the gatekeeper and its identifier.

Automatic Gatekeeper Configuration

To configure automatic gatekeeper discovery:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

5. auto-gk-discovery—Enable this parameter to use automatic gatekeeper discovery. The default value is disabled. Valid values are:

   - enabled | disabled

6. multicast—Set this parameter to the address and port where the RAS Multicast Group listens. Your entries in this field will be comprised of an IPv4 address and port values separated by a colon. The default value is 0.0.0.0:0.

H.323 Alternate Routing

You can configure your Oracle Enterprise Session Border Controller to try more possible routes within given time constraints and number of retries.

Without Alternate Routing Enabled

If you do not enable H.323 alternate routing, the Oracle Enterprise Session Border Controller tries one possible next hop gateway when routing H.323 calls even if the applicable local policy has multiple next hops configured. If that next hop gateway fails (either because it is busy or out of service), the Oracle Enterprise Session Border Controller relays the failure back to the caller, who hears a busy tone.

In addition, the call will only be routed to the other available next hops if the first one is:

- A session agent that has gone out of service because it constraints have been exceeded
- A session agent that has gone out of service because it failed to respond to a Oracle Enterprise Session Border Controller Setup request
- A session agent group

With Alternate Routing Enabled

When you enable H.323 Alternate Routing on your Oracle Enterprise Session Border Controller, you enable the use of the other next hops in addition to the first one. The retry, when the other available next hops are used, is transparent to the caller. However, the number of retries is limited by the value you set for the ACLI connect-tmo parameter, and this feature works only if there is more than one matching local policy next hop. If there is not more than one match, even if that match is a session agent group, then the call is only attempted once and the caller must retry it.

If the Oracle Enterprise Session Border Controller receives a Release Complete message before it receives an Alerting message, then it will try the next hop if there are multiple matches. When there is no more than one match, or if the timer or number of retries is exceeded, the Oracle Enterprise Session Border Controller proxies the most recently received Release Complete message back to the caller.

The following table shows the cause codes and release complete reasons, and either of the two actions the Oracle Enterprise Session Border Controller takes:

- Recur—Means that the Oracle Enterprise Session Border Controller performs (or continues to perform) alternate routing
- Proxy—Means that alternate routing stops, and the Oracle Enterprise Session Border Controller sends a release complete message back to the caller

H.323 Release Complete Reason	Q.850 Cause Code	Action
No Bandwidth	34—No circuit available	Recur
Gatekeeper Resources	47—Resource unavailable	Recur
Unreachable Destination	3—No route to destination	Recur
Destination Rejection	16—Normal call clearing	Proxy
Invalid Revision	88—Incompatible destination	Recur
No Permission	111—Interworking, unspecified	Recur
Unreachable Gatekeeper	38—Network out of order	Recur
Gateway Resources	42—Switching equipment congestion	Recur
Bad Format Address	28—Invalid number format	Recur
Adaptive Busy	41—Temporary Failure	Recur

H.323 Release Complete Reason	Q.850 Cause Code	Action
In Conference	17—User busy	Proxy
Undefined Reason	31—Normal, unspecified	Recur
Facility Call Deflection	16—Normal, call clearing	Proxy
Security Denied	31—Normal, unspecified	Recur
Called Party Not Registered	20—Subscriber absent	Recur
Caller Not Registered	31—Normal, unspecified	Recur
New Connection Needed	47—Resource Unavailable	Recur
Non Standard Reason	127—Interworking, unspecified	Recur
Replace With Conference Invite	31—Normal, unspecified	Recur
Generic Data Reason	31—Normal, unspecified	Recur
Needed Feature Not Supported	31—Normal, unspecified	Recur
Tunnelled Signaling Rejected	127—Interworking, unspecified	Recur

H.323 Alternate Routing Configuration

This section describes how to enable H.323 alternate routing. There is a new parameter, and the behavior of the pre-existing response-tmo and connect-tmo parameters change when you enable this feature on your system.

To enable this feature, you need to set the new alternate-routing parameter in the global H.323 configuration to recur. The other option for this parameter is proxy, which means that the Oracle Enterprise Session Border Controller performs in the way it did prior to Release 4.1, i.e. try only the first matching local policy next hop that it finds.

You configure H.323 alternate for the global H.323 configuration.

To enable H.323 alternate routing:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. alternate-routing—Enable or disable H.323 alternate routing. If you want to keep the pre-4.1 behavior where the Oracle Enterprise Session Border Controller only tries one matching local policy next hop, leave this parameter set to its default value proxy. Valid values are:

   - recur | proxy

5. response-tmo—Enter the time in seconds for the response time-out (or T303 timer). This is the amount of time allowed to elapse during which the Oracle Enterprise Session Border Controller should receive a response to its Setup message. If the first response to the Oracle Enterprise Session Border Controller's Setup is a callProceeding, then the Oracle Enterprise Session Border Controller should receive an Alerting or Connect message before this timer (now T303*2) elapses.

   The default for this parameter is 4. The valid range is:

   - Minimum—0
   - Maximum—999999999

6. connect-tmo—Enter the time in seconds for the connect time-out (or T301 timer). This is the amount of time allowed to elapse during which the Oracle Enterprise Session Border Controller should receive a Connect message.

   For alternate routing, this parameter is also used to limit the number of next hops that are tried and the length of time they are tried in case the first next hop fails. The call needs to be established before this timer expires; the call will fail after maximum of 5 retries.

   The default for this parameter is 32.

   - Minimum—0
   - Maximum—999999999

H.323 LRQ Alternate Routing

There are networks where the Oracle Enterprise Session Border Controller is positioned so that it needs to send an H.225 LRQ request to one signaling entity, and then fall back to another signaling entity when there are no resources available on the first. This might be the case when network contain elements that have limited amounts of channels and/or ports.

To handle situations like this one, the Oracle Enterprise Session Border Controller can be configured for H.323 LRQ alternate routing.

Without this feature enabled, the Oracle Enterprise Session Border Controller performs H.323 alternate routing for an H.323 call by finding the alternate route for a local policy when the call setup using H.225/Q.931 fails. Some network configurations, however, require that an LRQ message be sent to a gatekeeper prior to call setup in order to request the destination call signaling address—meaning that the Oracle Enterprise Session Border Controller will release the call if it does not receive an LCF for that LRQ.

With H.323 LRQ alternate routing enabled, the Oracle Enterprise Session Border Controller can route the call even when it does not receive the LCF.

When the Oracle Enterprise Session Border Controller routes an H.323 call using a local policy and the applicable route specifies gatekeeper/session agent as the next hop, the Oracle Enterprise Session Border Controller must send that gatekeeper an LRQ to request the destination for the call signaling address. After it sends the LRQ, the Oracle Enterprise Session Border Controller might receive either an LCF or an LRJ, or it might receive no response at all. Upon failure—either the receipt of an LRJ or no response within a timeout period—the Oracle Enterprise Session Border Controller tries alternate routes (additional routing policies) until the call is either set up or the routing list ends. For each alternate route, if the next hop is a gatekeeper/session agent, the Oracle Enterprise Session Border Controller sends an LRQ to the gatekeeper in order to request the destination call signaling address. Otherwise, the Oracle Enterprise Session Border Controller simply sets up the call.

For a designated period of time, the Oracle Enterprise Session Border Controller waits for the a response to the LRQ from the gatekeeper. This timeout period is configured by setting two options in the global H.323 configuration: ras-tmo (number of seconds the Oracle Enterprise Session Border Controller waits before retransmitting a RAS message; default is 4) and maxRasRetries (maximum number of times the Oracle Enterprise Session Border Controller retransmits the RAS; default is 1). The Oracle Enterprise Session Border Controller calculates the LRQ timeout period by multiplying the ras-tmo by the maxRasRetries and adding one (ras-tmo x maxRasRetries +1).

If an out of service session agent is part of a route, the Oracle Enterprise Session Border Controller skips it when using alternate routing and uses other routes for the policy.

A session agent might go out of service when it exceeds the maximum number of consecutive transaction timeouts to the maximum number of allowable transaction timeouts. Applicable session agent constrain parameter of note are:

- trans-timeouts—Maximum number of allowable transaction timeouts (default is 5)
- ttr-no-response—Dictates when the SA (Session Agent) should be put back in service after the SA is taken OOS (Out Of Service) because it did not respond to the Oracle Enterprise Session Border Controller
- in-service-period—Amount of time that elapses before a session agent is put back in service after the ttr-no-response period has passed

By default, the Oracle Enterprise Session Border Controller continues to send LRQ messages to a session agent even if the session agent has already sent an LRJ. However, you might want to place a session agent out of service when it has sent a certain number of LRJs; doing so allows alternate routing to take place faster, but this is an optional feature.

To configure an LRJ threshold, you add the max-lrj value to an H.323 session agent's options parameter; instructions for how to set it and the required syntax appear below. If you do not set this option, then the Oracle Enterprise Session Border Controller will not put session agents out of service for matters related to LRJs.

If you do set this option (to a non-zero value), then the Oracle Enterprise Session Border Controller keeps a count of the LRJs received from a session agent. When it receives an LCF from a session agent, the Oracle Enterprise Session Border Controller resets the counter to zero. This count is used internally only and is not accessible through statistics displays.

If a session agent exceeds the maximum number of LRJs and goes out of service, it remains in that state until the ttr-no-response period has passed and it has transitioned through the in-service-period time. If the ttr-no-response period is zero, then the session agent is never put out of service.

Caveats

The Oracle Enterprise Session Border Controller does not support H.323 LRQ alternate routing for these scenarios:

- Calls that require translation between SIP and H.323 (IWF calls)
- For pure H.323 calls where the ingress H.323 interface (stack) is associated with another H.323 interface (stack) that has a valid gatekeeper defined; if there is no valid gatekeeper for the egress interface (stack), this feature may apply.

H.323 LRQ RAS Retransmission Configuration

There is no configuration for H.323 LRQ alternate routing; it is enabled by default. You do, however, need to set the ras-tmo and maxRasRetries options to set the timeout period.

If you want to set a maximum number of consecutive LRJs to be received from a session agent, you need to add the max-lrj value to an H.323 session agent's options parameter.

To configure the number of seconds before the Oracle Enterprise Session Border Controller retransmits a RAS message:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ACMEPACKET(h323)#
   ```

4. options—Set the options parameter by typing options, a Space, the option name ras-tmo=x (where X is number of the seconds that the Oracle Enterprise Session Border Controller waits before retransmitting a RAS message; default is 4) with a plus sign in front of it, and then press Enter.

Set the maxRasRetries option in the same way; here, X is the maximum number of times the Oracle Enterprise Session Border Controller retransmits the RAS; default is 1).

```
ACMEPACKET(h323-stack)# options +ras-tmo=6
ACMEPACKET(h323-stack)# options +maxRasRetries=2
```

If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to the h323 configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

H.323 LRJ Limit Configuration

To limit the number of LRJs received from an H.323 session agent before putting it out of service:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
```

4. Use the ACLI select command so that you can work with the session agent configuration to which you want to add this option.

```
ACMEPACKET(session-agent)# select
```

5. options—Set the options parameter by typing options, a Space, the option name max-lrj=X (where X is the maximum number of allowed LRJs) with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(session-agent)# options +max-lrj=3
```

   If you type options max-lrj=X (without the plus sign), you will overwrite any previously configured options. In order to append the new option to the session-agent's options list, you must prepend the new option with a plus sign as shown in the previous example.

H.323 CAC Release Mechanism

When an OLC message is sent to the Oracle Enterprise Session Border Controller and there is insufficient bandwidth available, the Oracle Enterprise Session Border Controller will reject the incoming OLC. Normally, endpoints decide whether they want to send new OLCs or if they want to release the call. Some endpoints in this situation do neither. When communicating with the last of endpoints, it is desirable for the Oracle Enterprise Session Border Controller to take action.

The system supports a option in the H.323 interface called olcRejectTimer. When this option is enabled and an OLC is rejected, the stack will:

- If there is another media channel open, the Oracle Enterprise Session Border Controller will behave as if the release mechanism had not been enabled
- If there are no media channels open, the Oracle Enterprise Session Border Controller starts a timer for 1 second.
  - If the call is released by the endpoint before the timer expires or another OLC is received from the endpoint before the timer expires, the Oracle Enterprise Session Border Controller stops the timer and follows expected call handling
  - If the timer expires before either of the above responses from the endpoint occur, the Oracle Enterprise Session Border Controller releases the call.

H.323 CAC Release Configuration

To enable the H.323 CAC release mechanism:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

5. Use the ACLI select command so can add this feature to an existing H.323 interface.

```
ACMEPACKET(h323-stacks)# select
```

6. Set the options parameter by typing options, a Space, the option name olcRejectTimer, and then press Enter.

   ACMEPACKET(h323-stacks)# options olcRejectTimer

7. If you are adding this service to an H.323 interface that already exists, type select to select the interface to which you want to add the service. Then use the options command and prepend the option with a plus (+) sign.

   - If you know the same of the interface, you can type the name of the interface at the name: prompt and press Enter.
   - If you do not know the name of the interface, press Enter at the name: prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press Enter.
   - If are adding service to an existing interface and type in the option without a plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +olcRejectTimer.

H.323 Per-Realm CAC

Building on the Oracle Enterprise Session Border Controller's pre-existing call admission control methods, CAC can be performed based on how many minutes are being used by SIP or H.323 calls per-realm for a calendar month.

In the realm configuration, you can now set a value representing the maximum number of minutes to use for SIP and H.323 session using that realm. Although the value you configure is in minutes, the Oracle Enterprise Session Border Controller performs CAC based on this value to the second. When you use this feature for configurations with nested realms, the parent realm will have the total minutes for all its child realms (i.e., at least the sum of minutes configured for the child realms).

The Oracle Enterprise Session Border Controller calculates the number of minutes used when a call completes, and counts both call legs for a call that uses the same realm for ingress and egress. The total time attributed to a call is the amount of time between connection (H.323 Connect) and disconnect (H.323 Release Complete), regardless of whether media is released or not; there is no pause for calls being placed on hold.

If the number of minutes is exhausted, the Oracle Enterprise Session Border Controller rejects calls with a SIP 503 Service Unavailable message (including additional information "monthly minutes exceeded). In the event that the limit is reached mid-call, the Oracle Enterprise Session Border Controller continues with the call that pushed the realm over its threshold but does not accept new calls. When the limit is exceeded, the Oracle Enterprise Session Border Controller issues an alarm and sends out a trap including the name of the realm; a trap is also sent when the alarm condition clears.

Note: The Oracle Enterprise Session Border Controller does not reject GETS/NSEP calls based on monthly minutes CAC.

You can change the value for minutes-based CAC in a realm configuration at any time, though revising the value downward might cause limits to be reached. This value resets to zero (0) at the beginning of every month, and is checkpointed across both system in an HA node. Because this data changes so rapidly, however, the value will not persist across and HA node if both systems undergo simultaneous failure or reboot.

You can use the ACLI show monthly minutes <realm-id> command (where <realm-id> is the realm identifier of the specific realm for which you want data) to see how many minutes are configured for a realm, how many of those are still available, and how many calls have been rejected due to exceeding the limit.

Caveats

Note: this feature is not supported for HA nodes running H.323.

H.323 Per-Realm CAC Configuration

This section shows you how to configure minutes-based CAC for realms and how to display minutes-based CAC data for a specific realm.

Note that setting the new monthly-minutes parameters to zero (0), or leaving it set to its default of 0, disables this feature.

To configure minutes-based CAC:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
Oracle Enterprise Session Border Controller(configure)# media-manager
Oracle Enterprise Session Border Controller(media-manager)#
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to want to add SIP per user CAC.

```
ACMEPACKET(realm-config)# select
```

5. monthly-minutes—Enter the number of minutes allowed during a calendar month in this realm for SIP and H.323 calls. By default, this parameter is set to zero (0), which disabled monthly minutes-based CAC. You can enter a value as high as 71582788.
6. Save and activate your configuration.

Use the ACLI show monthly-minutes command to see the following information:

- How many minutes are configured for a realm
- How many of those are still available
- How many calls have been rejected due to exceeding the limit

To view information about SIP per user CAC using the IP address mode: In either User or Superuser mode, type show monthly-minutes <realm-id>, a Space, and the IP address for which you want to view data. Then press Enter. The <realm-id> is the realm identifier for the realm identifier of the specific realm for which you want data.

```
ACMEPACKET# show monthly-minutes private_realm
```

H.323 Bearer-Independent Setup

In Release 4.1, the Oracle Enterprise Session Border Controller supports a new H.323 option that enables H.323 Bearer-Independent Setup (BIS). When enabled, this feature allows exception to slow-start to fast-start conversion on the Oracle Enterprise Session Border Controller.

H.323 BIS Disabled

Unless you enable this feature, the Oracle Enterprise Session Border Controller performs slow-start to fast-start conversion when a call entering the system as slow-start was routed to a an outgoing H.323 interface (stack) with call-fast-start set to enabled and there is a list of valid media-profiles in the configuration.

H.323 BIS Enabled

There are certain cases in access deployments where the slow-start to fast-start conversion should not be applied. This is the case when the Setup message contains the Bearer Capability information element (IE), which signals BIS.

When you enable this feature and the Oracle Enterprise Session Border Controller receives an incoming Setup message that does not contain a fastStart field, the Oracle Enterprise Session Border Controller checks for the BIS in the incoming Setup before it starts to perform the slow-start to fast-start conversion. If it finds the BIS, then it does not perform the conversion.

This feature can be enabled on a global or a per-interface basis, meaning that you can apply it to your system's entire H.323 configuration or you can enable it only for the interfaces where you want it applied.

H.323 BIS Global Configuration

This section explains how to add H.323 BIS support to your global H.323 configuration and to specific H.323 interfaces (stacks).

If you set this option on an H.323 interface (stack), you must set it on the interface (stack) that receives the Setup message with BIS in the Bearer Capability IE.

To enable the H.323 BIS feature globally:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type options +bearerIndSetup and press Enter.

   ```
   ACMEPACKET(h323-stacks)# options +bearerIndSetup
   ```

   If you type options bearerIndSetup without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

H.323 BIS Specific Configuration

To enable the H.323 BIS feature for a specific H.323 interface:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ```

4. Type h323-stacks and press Enter.

   ```
   ACMEPACKET(h323)# h323-stacks
   ACMEPACKET(h323-stacks)#
   ```

5. Select the H.323 stack to which you want to add H.323 BIS support.

   ```
   ACMEPACKET(h323-stacks)# select
   <name>:
   ```

   For a list of configured H.323 interfaces (stacks), press Enter at the <name>: prompt. Then enter the number corresponding to the interface where you want to apply this feature.

6. Type options +bearerIndSetup and press Enter.

   ```
   ACMEPACKET(h323-stacks)# options +bearerIndSetup
   ```

   If you type options bearerIndSetup without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

TOS Marking for H.323 Signaling

You can configure your Oracle Enterprise Session Border Controller to perform TOS/DiffServ marking for H.323 signaling packets. This feature enables you to mark H.323 signaling packets so that they receive specific treatment from upstream devices. This feature assists in routing because you can configure the TOS byte inserted in the H.323

packet to mark the traffic for certain destinations. For example, you can prevent unauthorized video transmission through an audio-only session.

The Oracle Enterprise Session Border Controller also performs TOS/DiffServ marking for media.

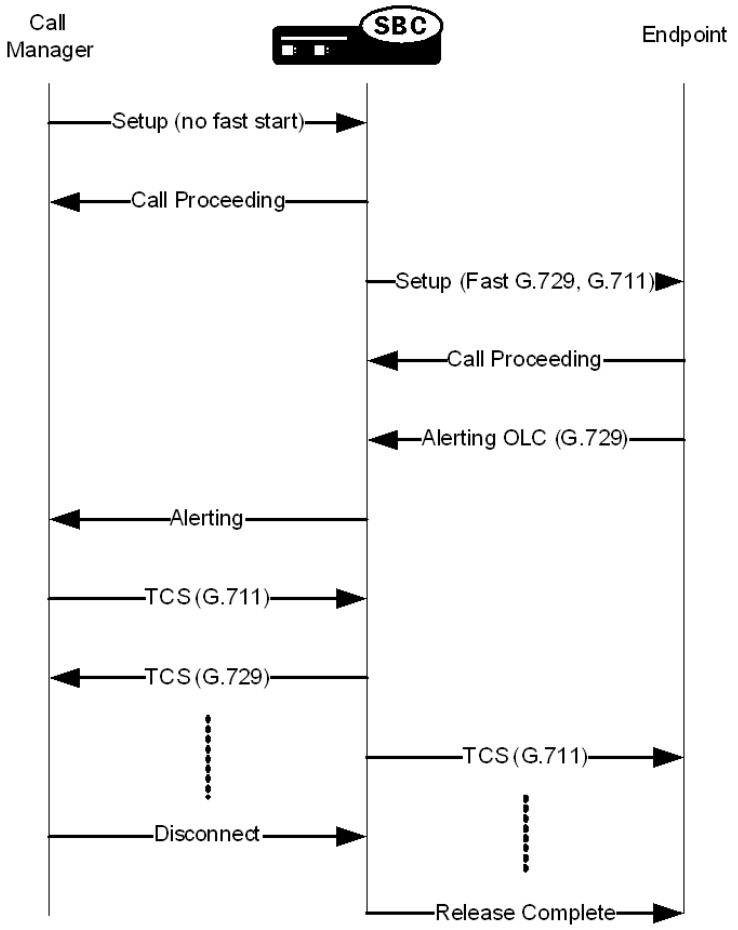
H.323 Codec Fallback

In the global H.323 configuration, you can enable a parameter that allows the Oracle Enterprise Session Border Controller to renegotiate—or fallback—to the preferred codec used in an incoming terminal capability set (TCS) from the slow-start side of a slow-start to fast-start H.323 call. When enabled, the Oracle Enterprise Session Border Controller performs this renegotiation when it detects a mismatch between the codec used in the open logical channel (OLC) opened on the fast-start side of the call, and the codec specified by the slow-start side.

Codec Fallback Disabled

With codec fallback disabled, the Oracle Enterprise Session Border Controller opens a channel using the codec specified by the northbound side. Since the call manager had specified another preferred codec, the result is a codec mismatch leading to a dropped call.

The following diagram shows how codec mismatches end in dropped calls.

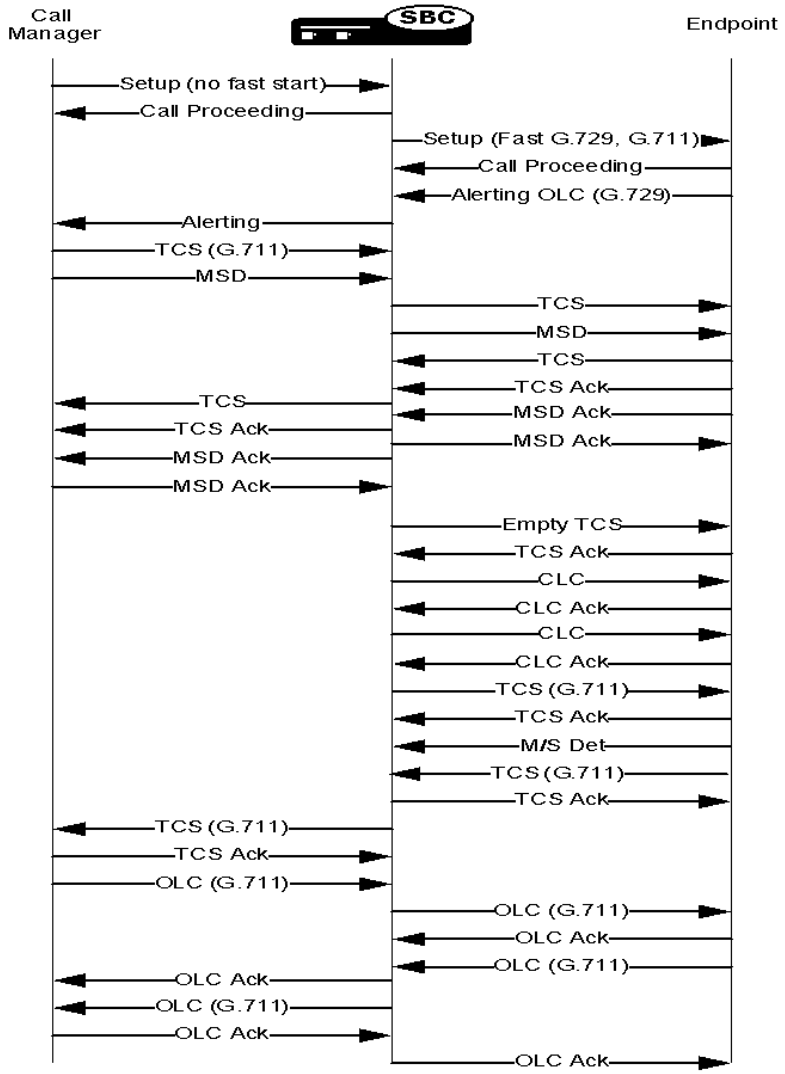


Codec Fallback Enabled

With H.323 codec fall back enabled, the Oracle Enterprise Session Border Controller attempts to use the preferred codec that the slow-start side of the call specifies. The Oracle Enterprise Session Border Controller determines matching based on the incoming TCS from the slow-start side and the OLC on the egress side. If the codecs do not match, the Oracle Enterprise Session Border Controller sends an empty TCS on the egress side and closes the logical channels on the outgoing side of the call.

To trigger a new capabilities exchange, the Oracle Enterprise Session Border Controller forwards the TCS from the ingress side of the call to the egress endpoint. Then the TCS from the egress endpoint is propagated to the ingress endpoint, and the logical channels are opened.

The following diagram shows a call scenario using the H.323 codec fallback feature.



H.323 Codec Fallback Configuration

Note that you configure this feature for your global H.323 configuration, so it has an impact on all H.323 traffic on your system.

To enable H.323 codec fallback:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter. The system prompt will change to let you know that you can configure individual

   ```
   ACMEPACKET(session-router)# h323
   ```

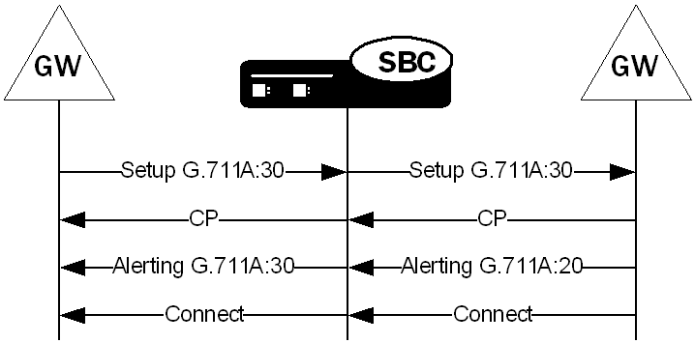
4. codec-fallback—Enable or disable the H.323 codec fallback feature. The default value is disabled. Valid values are:

- enabled | disabled

H.323 TCS Media Sample Size Preservation

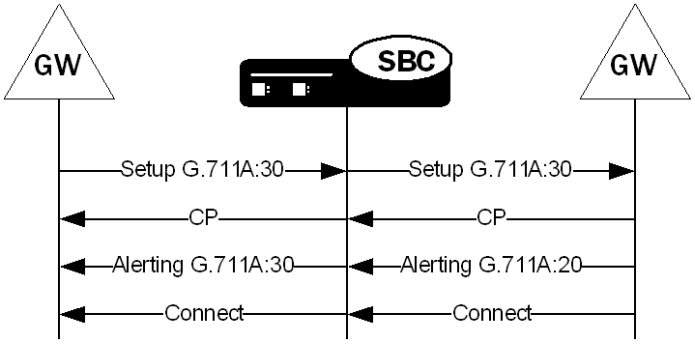
For H.323 fastStart calls, the Oracle Enterprise Session Border Controller can be configured to preserve the packetization interval from the called gateway if it differs from the one offered in the Setup message the calling gateway sent.

When this feature is disabled and in accordance with the ITU H.323 recommendation, the Oracle Enterprise Session Border Controller changes the packetization rate to the one used by the calling gateway if the one offered by the called gateway differs. In the following example, this means that the Oracle Enterprise Session Border Controller replaces the packetization interval of 20 with 30 before it forwards the Alerting message to the calling gateway.



However, not all H.323 elements comply with the ITU recommendation. Since some network elements do modify the packetization rate in the dataType element, this behavior is now configurable.

When you enable media sample size preservation, the Oracle Enterprise Session Border Controller allows the packetization rate to be modified and forwards on the modified dataType element to the calling gateway. In the following example, you can see that the Oracle Enterprise Session Border Controller forwards the called gateway's Alerting with the packetization interval of 20 despite the fact that the calling gateway's Setup specified 30.



Note that the calling endpoint might or might not work with the modified dataType.

You can enable this feature for the global H.323 configuration so that it applies to all H.323 fastStart calls, or you can enable it on a per-H.323 interface (stack) basis. When you enable this feature for an individual H.323 interface (stack), the Oracle Enterprise Session Border Controller performs media sample size preservation for calls egressing on that interface.

Media Sample Size Configuration

This section shows you how to configure media sample size preservation for the global H.323 configuration and for an individual H.323 interface (stack).

To enable media sample size preservation for the global H.323 configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. options—Set the options parameter by typing options, a Space, the option name forwardFSAceptedDataType with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(h323)# options +forwardFSAcceptedDataType
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to the h323 configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

   To enable media sample size preservation for an individual H.323 interface:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

8. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

9. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

   If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI select command) the one you want to edit.

10. options—Set the options parameter by typing options, a Space, the option name forwardFSAceptedDataType with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(h323-stack)# options +forwardFSAcceptedDataType
```

    If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to the h323-stack configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

11. Save and activate your configuration.

H.323-TCS H.245 Support for H.264 and G722.1

The Oracle Enterprise Session Border Controller supports the H.264 video codec and the G722.1 audio codec. Especially useful for customer video product offerings in which the Oracle Enterprise Session Border Controller is deployed, this support further allows the Oracle Enterprise Session Border Controller to increase ease of use by supporting private addressing. Without this feature enabled (the Oracle Enterprise Session Border Controller's previous behavior), the Oracle Enterprise Session Border Controller required deployment for IANA registered IP addresses—despite the fact that IP VPNs allow for RFC 1918 private addressing.

H.323-TCS Generic Video Configuration

To enable this feature, you need to set up media profile configurations appropriately. Media profiles now allow you to set the configuration either as "generic video" or generic audio.

H.245 provides for defining new capabilities that are described as H.245 generic capabilities (GenericCapability), which the Oracle Enterprise Session Border Controller now supports using the H.245 GenericCapability structure. H. 264 and G.722.1 are the first codecs the Oracle Enterprise Session Border Controller offers that use this mechanism.

To set a media profile for generic video support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. name—Set the name of the generic video media profile to genericVideo. There is no default for this parameter.
5. media-type—Set the media type to use for this media profile; for generic video, set this parameter to video.
6. payload-type—Set the payload type to use for the generic video media profile.
7. transport—Set the transport type to use for the generic video media profile.
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic video media profile configuration:

```
media-profile
        name                            genericVideo
        media-type                     video
        payload-type                    99
        transport                       RTP/AVP
        req-bandwidth                   0
        frames-per-packet               0
        parameters
        average-rate-limit              0
        sdp-rate-limit-headroom         0
        sdp-bandwidth                   disabled
```

H.323-TCS Generic Audio Configuration

To set a media profile for generic audio support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. name—Set the name of the generic audio media profile to genericAudio. There is no default for this parameter.
5. media-type—Set the media type to use for this media profile; for generic video, set this parameter to audio.
6. payload-type—Enter the format in SDP m lines. No payload type number is assigned for newer, dynamic codecs. For RTP/AVP media-profile elements, this field should only be configured when there is a standard payload type number that corresponds to the encoding name. Otherwise, this field should be left blank. This field is used by the system to determine the encoding type when the SDP included with a session identifies the standard payload type on the em line, but does not include an a-rtpmap entry.
7. transport—Set the type of transport protocol to use for the generic audio media profile. The default value is RTP/AVP.
   - UPD | RTP/AVP
8. Complete the rest of the media profile configuration as needed.

9. Save and activate your configuration.

   The following is a sample of a generic audio media profile configuration:

```
media-profile
        name                            genericAudio
        media-type                      audio
        payload-type                    104
        transport                       RTP/AVP
        req-bandwidth                   0
        frames-per-packet               0
        parameters
        average-rate-limit              0
        sdp-rate-limit-headroom         0
        sdp-bandwidth                   disabled
```

International Peering with IWF and H.323 Calls

When you do not enable this feature, H.323 calls can default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option cpnType=X, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outing calls on the h323-stack.

The following is a list of possible cpnType=X option values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

You configure this feature as an option in the h323-stack configuration.

To configure the cpnType=X option for H323-H323 calls:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323)#
```

4. Type h323-stacks and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

5. Set the options parameter by typing options, a Space, the option name cpnType=x with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(h323-stack)# options +cpnType=x
```

   If you type options without the plus sign, you will overwrite any previously configured options. In order to append the new options to the h323-stack's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

Default OLC Behavior Changed in Upgrade

You can configure the Oracle Enterprise Session Border Controller to force media profiles in OLC messages when negotiating H.323 calls. This is the default behavior prior to Release S-C6.1.0.

In Release 6.1.0 and forward the default behavior of OLC is to inherit the values recieved in the signaling message from the remote endpoint.

No license requirements to enable this feature.

To enable forced media profiles in OLC:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323-config and press Enter.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

4. options—Set the options parameter by typing options, a Space, the option-name forceMediaProfileInOLC with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(h323-config)# options +forceMediaProfileInOLC
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the SIP interface configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Options

The options parameter in the global H.323 and H.323 interface configurations allows you to establish the use of specific features; most of those features are customer specific.

You should exercise caution when you apply options because of the fact that many of them are for customer-specific applications. Consult with your Acme Packet systems engineering to find out if using a particular option would be an advantage to you.

Under no circumstance do we recommend that you configure options without Oracle consultation. There is the chance that you could set an option that might harm an otherwise sound configuration.

Some of the options described below are only applicable to IWF calls. However, you need to establish them in your H.323 configuration.

Global H.323 Options

The following table lists the options that you might want to use in the global H.323 configuration. Again, we recommend that you consult with an Oracle systems engineer about your configuration before using any of these options.

Options	Description
NoDynamicMSD	Oracle Enterprise Session Border Controller forcefully assumes the "master" role for an outgoing call, and the slave role for an incoming call.
AllowOLCWoMSD	Oracle Enterprise Session Border Controller sends OLC before master/slave determination is complete.

Options	Description
	Causes the Oracle Enterprise Session Border Controller to be noncompliant with the H.323 recommendation, which does not permit an OLC to be sent prior to MSD completion.
ModifyMediaInAck	Oracle Enterprise Session Border Controller accepts and propagates changes to media presented in an OLC Ack. Applies only to Fast Start OLC/OLC Ack messages embedded in H.225/Q.931 messages during call setup. Causes Oracle Enterprise Session Border Controller to be noncompliant with the H.323 recommendation, which does not permit media characteristic to be specified in an OLC to be changed in an OLCAck.
MapG729	Oracle Enterprise Session Border Controller maps H.245 G.729 to SDP G.729 with Annex B and vice versa. Applicable only to IWF calls.
ColonG729	Oracle Enterprise Session Border Controller uses the : (colon) instead of the = (equal sign) in the media attribute line a=fmtp:18 annexb=yes/no when mapping H.245 G.729 or SDP G.729 with Annex B. Applicable only to IWF calls.
IwfLRQ	Oracle Enterprise Session Border Controller sends an INVITE (with no SDP) to a redirect server in response to an incoming LRQ received on an H.323 interface. If a 3xx message with a redirected contact header is returned, the Oracle Enterprise Session Border Controller will send an LCF in response to the LRQ. Otherwise, it will send an LRJ.
NoG729AnnexB	SDP received by the IWF with H.729 and no FMTP will be mapped to G.729 on the H.323 side of the call. Can also be set in the session agent options parameter.
sameT38Port	Oracle Enterprise Session Border Controller does not allocates separate ports for audio and T.38. Oracle Enterprise Session Border Controller will send the same audio port in the OLCAck that it sees in a request mode for T.38 and a new OLC for T.38.
pvtStats	Oracle Enterprise Session Border Controller includes program value tree (PVT) statistics in the show h323d display that are a sum of the PVT statistics for all H.323 interfaces. Used for debugging purposes.
strayARQTimer	Required the syntax "strayARQTimer=x," where x is the number of seconds the system waits before tearing down an unsuccessful call in the case of stray ARQs.
forceMediaProfileInOLC	Reverts to older OLC Behavior.

H.323 Interface Options

The following table lists the options that you might want to use in the configuration H.323 interfaces. Again, we recommend that you consult with an Oracle systems engineer about your configuration before using any of these options.

Option	Description
stackAliasWins	Oracle Enterprise Session Border Controller will replace the sourceAddress of the incoming Setup message with the terminal alias of the egress interface when copying the incoming sourceAddress to the outgoing Setup message.
uniqueRRQRASAddress	Oracle Enterprise Session Border Controller will generate unique rasAddress for each RRQ that it sends to a gatekeeper in response to an incoming RRQ received on an H.323 interface configured for process registration. The IP address will be the local-ip of the outgoing interface, so the port is the unique portion of the rasAddress.

Option	Description
nonV4AdditiveRRQ	Gatekeeper associated with the H.323 interface support additive registration even though it does not set the additiveRegistration field in the RRQ message. When sending in the additive mode, the H.323 interface only sends with the RRQ new terminal aliases that need to be registered. In non-additive mode, the interface sense all the terminal aliases that have been registered, plus the new aliases.
cachedTerimnalAlias	Oracle Enterprise Session Border Controller copies the terminal alias(es) of the registered endpoint to the asourceAddress field of the Setup message. Terminal alias(es) are changed after the system successfully processes an RRQ from the endpoint.
proxySrcInfo	Oracle Enterprise Session Border Controller copies the sourceInfo from the incoming Setup message to the outgoing Setup message. Otherwise, Oracle Enterprise Session Border Controller uses its own endpointType for the sourceInfo field.
noAliasinRCF	Oracle Enterprise Session Border Controller does not include any terminal alias in the RCF.
forceH245	Oracle Enterprise Session Border Controllerinitiates an H.245 connection after the call is connected. Otherwise, Oracle Enterprise Session Border Controller listens for an H.245 connection to be initiated by a remote endpoint.
useCPNInRAS	Oracle Enterprise Session Border Controller uses the calling party number (CPN) IE of the incoming call as the srcInfo of a RAS message sent in the outgoing call (such as an ARQ).
maintenanceProxy	Oracle Enterprise Session Border Controller registers interfaces on the enterprise side with a gatekeeper on the carrier side, and registers endpoints through the Oracle Enterprise Session Border Controller with a unique rasAddress. Interfaces on the enterprise side are associated with the carrier interfaces; you set this option on the carrier side.
convertPNToE164	Oracle Enterprise Session Border Controller converts the address type partyNumber to dialedDigits (E.164). Conversion applies to sourceAddress, destinationAddress, and destExtraCallInfo aliases in Setup messages.
useCalledPNAsDestInfo	Oracle Enterprise Session Border Controller uses the H.225 called party number IE as the destinationInfo in ARQ and LRQ requests. Since translation rules can be applied to the Called Party Number, the option enables digit normalization for RAS requests. When not used, Oracle Enterprise Session Border Controller derives the destinationInfo field in RAS requests from the DestnationAddress field of the incoming Setup.
waitForIncomingH245	On the incoming leg, the Oracle Enterprise Session Border Controller does not send out its h245Address, but waits for the calling endpoint to send its H245Address.Applies to the outgoing call led as well: The Oracle Enterprise Session Border Controller does not send out a Facility with startH245 reason and waits for the called endpoint to send its H245Address.
uniqueRRQSrcPort	Enables H.323 RAS Port Mapping. The Oracle Enterprise Session Border Controller uses the RAS port that it assigned in the rasAddress parameters of an RRQ message as the UDP source port of the outgoing RRQ. Because this feature is linked to the unique RRQ functionality, be aware of the following before you enable the feature:

Option	Description
	• Enabling H.323 RAS Port Mapping automatically enables the Oracle Enterprise Session Border Controller's unique RRQ functionality, eliminating the need for you to configure the latter as a separate option. • Enabling the unique RRQ functionality (by setting the uniqueRRQRASAddress option) does not automatically enable H.323 RAS Port Mapping.
srcCallSignallingPort	Enables use of the Q.931 port value for the port field in the sourceCallSignalAddress parameter in an H.225 Setup message. Useful for customers who configure a separate H.323 interface (stack) on the core side for each external IP-PBX.

H.323 Stack Monitoring

In releases prior to S-C6.2.0, the Oracle Enterprise Session Border Controller provides SNMP monitoring of H.323 session agents but not of the H.323 stacks themselves. The H.323 stack/interface configuration now provides a way for you to set alarm thresholds on a per-stack basis. When enabled, this alarm system ties into the max-calls value to send critical, major, or minor alarms when the number of calls approaches the threshold.

Each H.323 stack now has a threshold crossing alert (TCA) where you can set up three severity levels: critical, major, and minor. You can define one severity level or all three for each stack. To prevent the alarm from firing continuously as call volume through the stack varies, each severity level has an has a reset value below the TCA you set. In addition, each threshold value resets when:

- An alarm with a higher severity is triggered, or
- The built-in reset value for the threshold level is 1% less than the parameter value

RTN 1477

H.323 Stack Monitoring Configuration

This section shows you how to configure H.323 stack monitoring for one H.323 stack configuration. This example shows one instance of the alarm-threshold sub-configuration being established; remember that you can set three—critical, major, and minor. Simply repeat the configuration steps to add more severity levels.

To set up H.323 stack monitoring:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323 and press Enter to access the global H.323 configuration.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. Type h323-stack and press Enter. If you are adding H.323 stack monitoring to an existing H.323 stack configuration, then remember you must select the stack you want to edit.

```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

5. Type alarm-threshold and press Enter to configure this feature.

```
ACMEPACKET(h323-stack)# alarm-threshold
ACMEPACKET(alarm-threshold)#
```

6. severity—Enter the type of severity level for the alarm you want to define. Choose from: critical, major, or minor. This value is required, and defaults to minor.

7. value—Enter the percentage of the number of calls defined in the max-calls parameter that triggers the alarm. For example, if you want to set a minor alarm to fire when the call rate through the stack reaches half the max-calls value, enter 50 (meaning 50%). The default value for this parameter is 0, which disables the alarm.

   Remember that if the number of calls falls to below 1% of the max-calls threshold you set, the clear trap fires.
8. Save your work. You can see the data related to this feature using the ACLI display-alarms and show h323 stack stack-alarms commands.

H.323 Automatic Features

This section describes H.323 features that are automatically enabled on your Oracle Enterprise Session Border Controller. You do not have to configure special parameters to turn them on. Even though you do not have to turn these features on, this section describes what they do and how they work.

Alias Mapping

Alias mapping permits destination addresses to be modified by a gatekeeper.

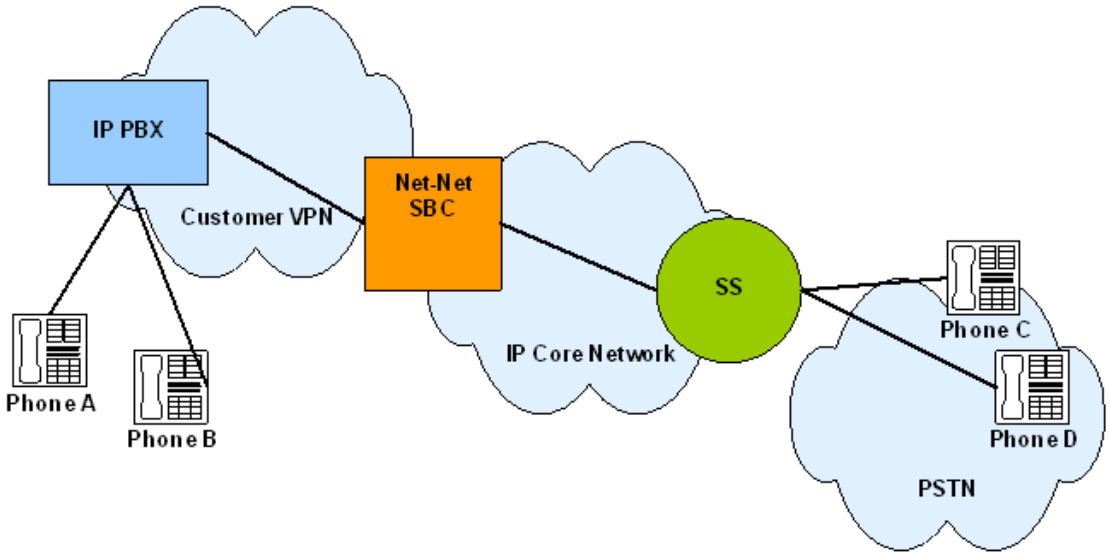
When sending an ARQ or an LRQ message to a gatekeeper, the Oracle Enterprise Session Border Controller sets the canMapAlias field in that message to true. This setting indicates that the Oracle Enterprise Session Border Controller accepts modified destination information from the gatekeeper. If the resulting ACF or LCF contains destinationInfo and/or destExtraCallInfo fields, then the Oracle Enterprise Session Border Controller copies that information respectively to the destinationAddress and destExtraCallInfo fields of the Setup message. In addition, if the destinationInfo is either type e164 or type partyNumber, the Oracle Enterprise Session Border Controller copies the information into the calledPartyNumber information element (IE) of the Setup message, replacing the existing calledPartyNumber IE.

You do not need to configure special parameters for this feature; it is enabled automatically.

Call Hold and Transfer

The Oracle Enterprise Session Border Controller's H.323 call hold and transfer feature supports consultation in addition to call holder and transfer. This feature uses signaling procedures based on the ITU-T recommendations/H. 323 specification for what it calls third party initiated pause and rerouting.

The following diagram shows how the Oracle Enterprise Session Border Controller is positioned to provide call hold and transfer support for H.323.



Call Hold and Transfer Basic Call

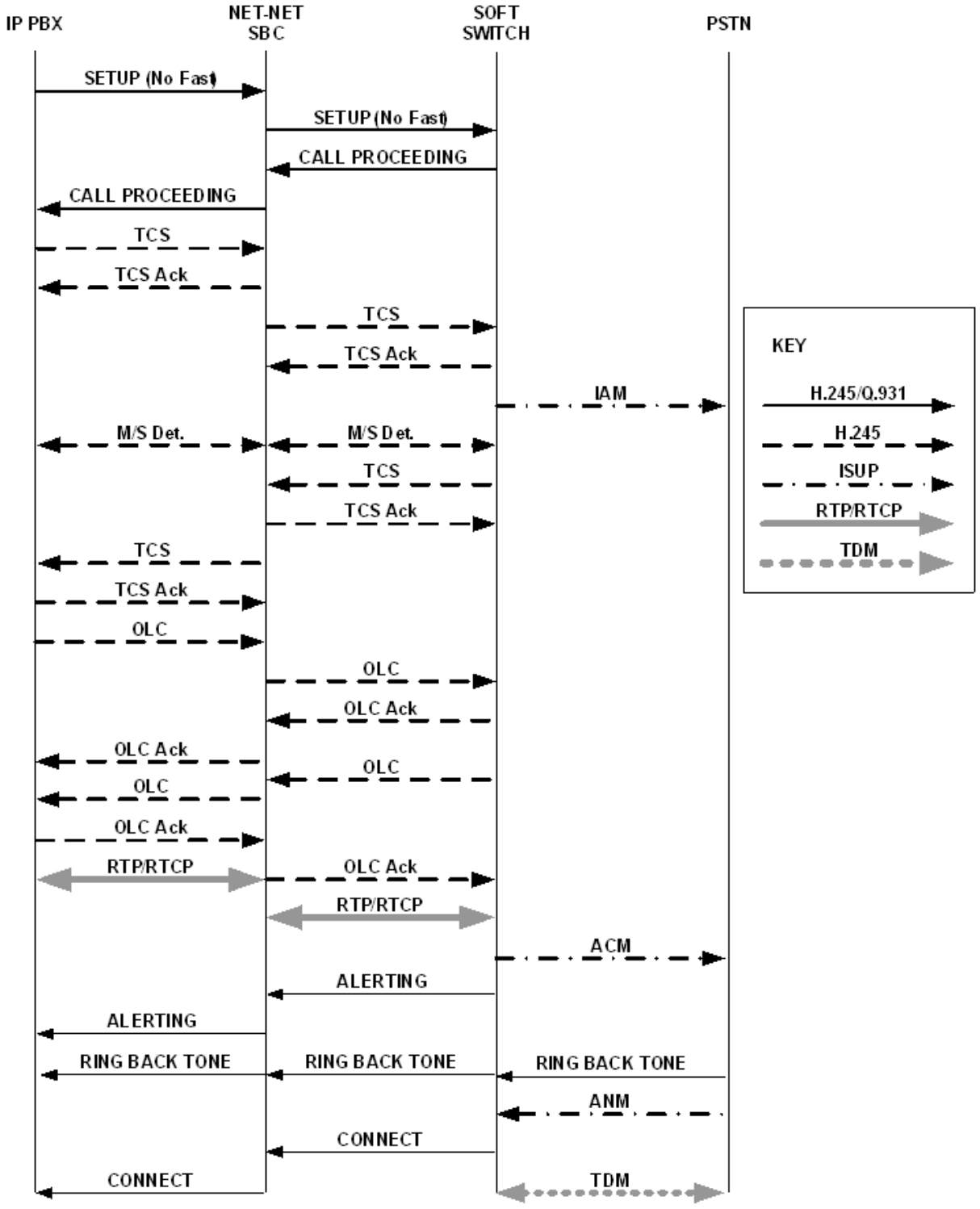
The following diagram show the signaling and media flows between the IP PBX and a softswitch. Note how the Oracle Enterprise Session Border Controller is position to mediate flows between the two devices.

In the Call Proceeding messages forwarded to the IP PBX, the Oracle Enterprise Session Border Controller uses a non-zero value to ensure that the IP PBX initiates an H.245 session. A progress indicator does not need to be included if the H.245 address is present in any of the following message types: Alerting, Progress, or Connect.

After the Oracle Enterprise Session Border Controller receives a Call Proceeding message from the softswitch that contains the H.245 address, the Oracle Enterprise Session Border Controller sends another Call Proceeding with its own H.245 address.

In the following call flow, the softswitch generates message to the gateway. These messages are:

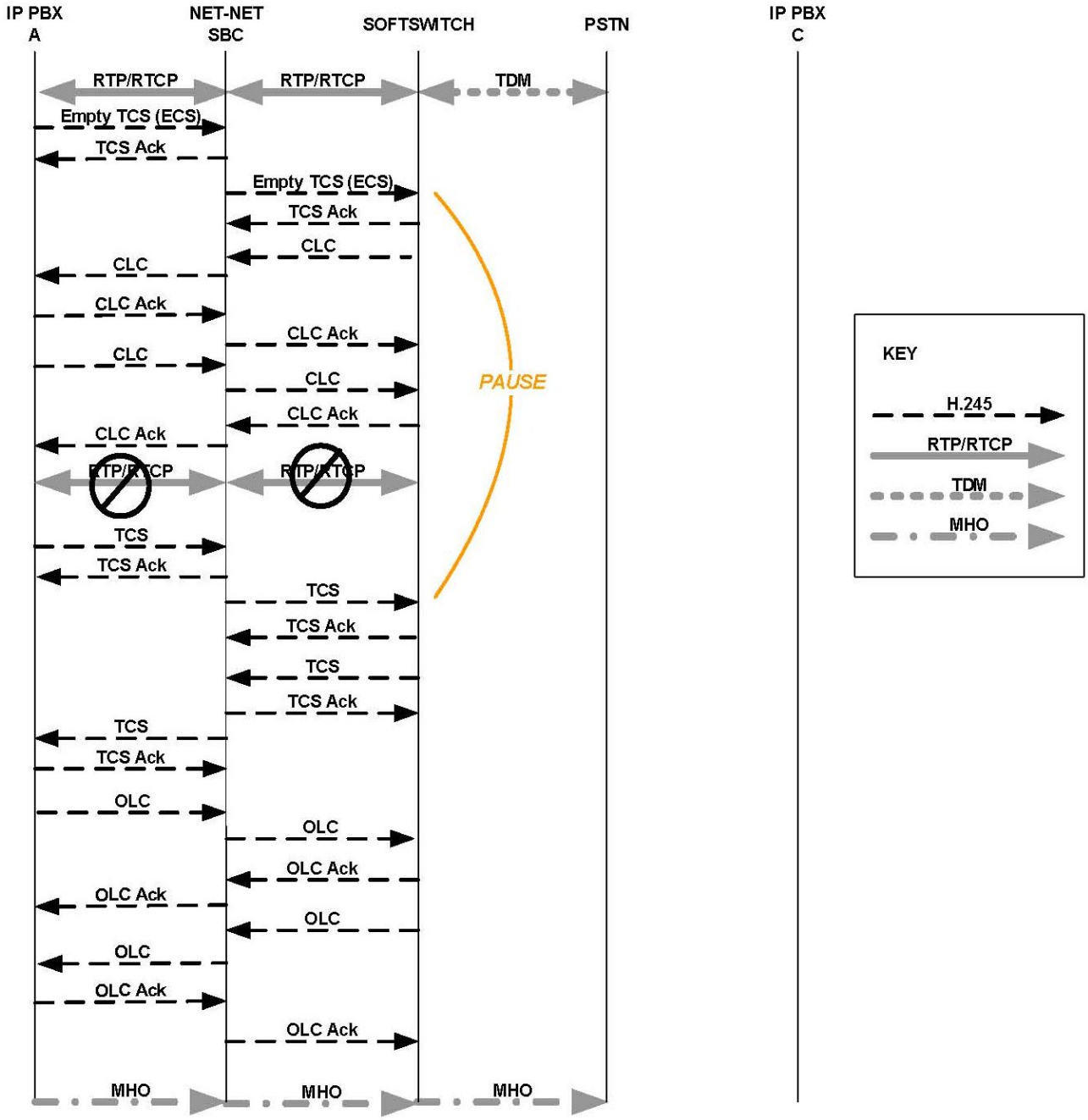
- Initial Address Message (IAM)
- Address Complete Message (ACM)
- Answer Message (ANM)



Call Hold and Transfer Music on Hold

The following diagram begins with the condition that IP PBX A is already connected with a gateway, with the Oracle Enterprise Session Border Controller and the softswitch positioned between the two.

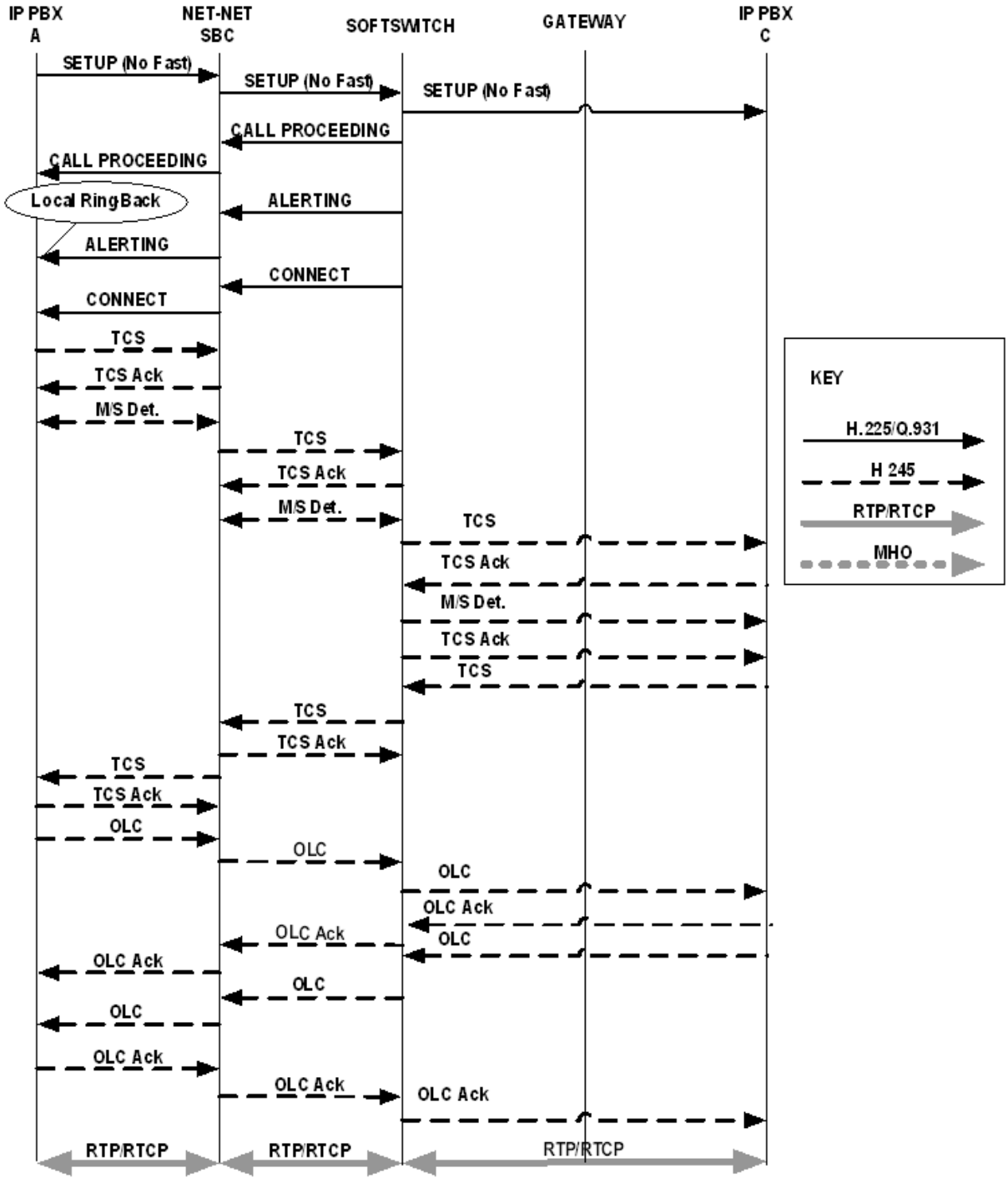
You can see in the call flow where the channels for transporting media are closed, and where the RTP/RTCP is stopped. This creates a pause for the call. With the Oracle Enterprise Session Border Controller mediating the process, IP PBX A and the softswitch exchange TCS and OLC messages that allow music on hold (MHO) to flow between IP PBX A and the gateway.

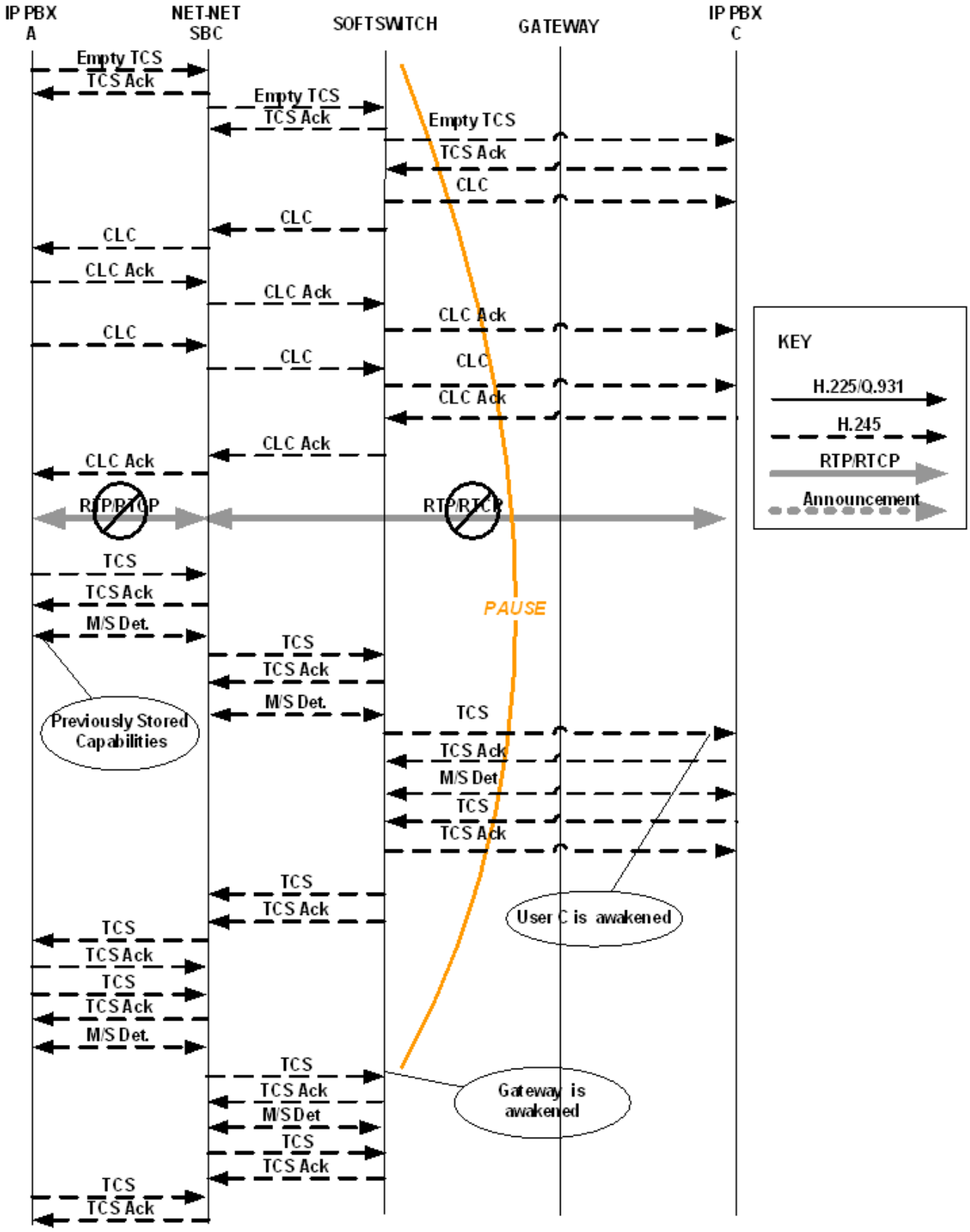


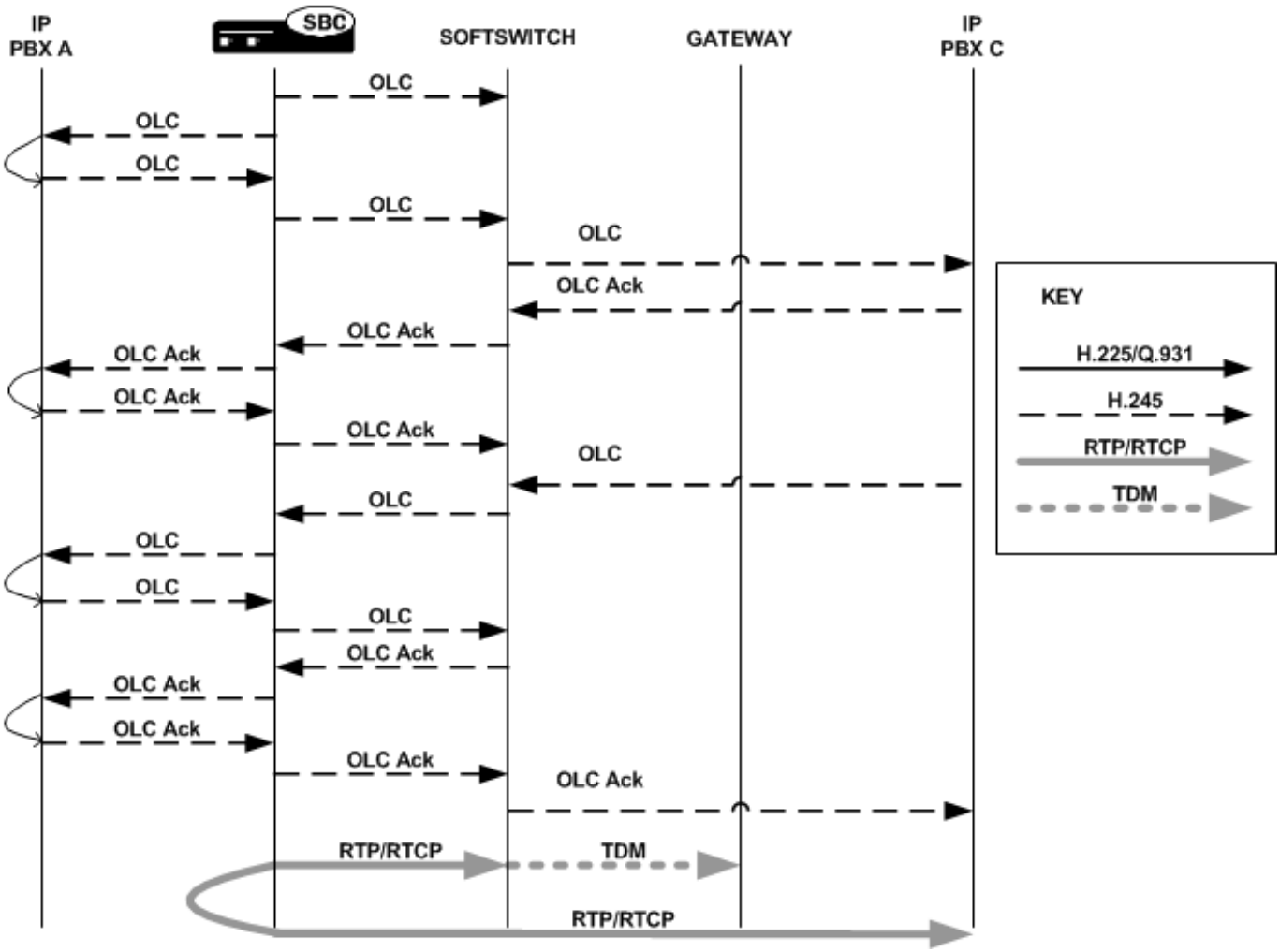
Call Hold and Transfer

The following diagram shows how call transfer works on the Oracle Enterprise Session Border Controller for H.323. In this diagram, you can see:

- Where local ringback occurs
- Where the pause begins and ends
- Where users and gateways are awakened
- Where logical channels are opened and closed


IP PBX A
NET-NET SBC
SOFTSWITCH
GATEWAY
IP PBX C
SETUP (No Fast)
SETUP (No Fast)
SETUP (No Fast)
CALL PROCEEDING
CALL PROCEEDING
Local RingBack
ALERTING
ALERTING
CONNECT
CONNECT
TCS
TCS Ack
M/S Det.
TCS
TCS Ack
M/S Det.
TCS
TCS Ack
M/S Det.
TCS Ack
TCS
TCS
TCS Ack
TCS
TCS Ack
OLC
OLC
OLC
OLC Ack
OLC
OLC Ack
OLC
OLC Ack
OLC
OLC Ack
OLC Ack
RTP/RTCP
RTP/RTCP
RTP/RTCP
KEY
H.225/Q.931
H 245
RTP/RTCP
MHO


IP PBX A
NET-NET SBC
SOFTSWITCH
GATEWAY
IP PBX C
Empty TCS
TCS Ack
Empty TCS
TCS Ack
Empty TCS
TCS Ack
CLC
CLC
CLC
CLC Ack
CLC Ack
CLC Ack
CLC
CLC
CLC
CLC Ack
CLC Ack
CLC Ack
RTP/RTCP
RTP/RTCP
TCS
TCS Ack
M/S Det.
PAUSE
TCS
TCS Ack
M/S Det.
Previously Stored Capabilities
TCS
TCS Ack
M/S Det
TCS
TCS Ack
User C is awakened
TCS
TCS Ack
TCS
TCS Ack
M/S Det.
TCS
TCS Ack
M/S Det
TCS
TCS Ack
Gateway is awakened
TCS
TCS Ack
KEY
H.225/Q.931
H.245
RTP/RTCP
Announcement



Media Release for SS-FS Calls

When the Oracle Enterprise Session Border Controller routes a slow-start to fast-start call, it is possible for the same fast-start call to be routed back through the Oracle Enterprise Session Border Controller making for a hairpin flow. If it does becomes a hairpin flow, then the Oracle Enterprise Session Border Controller routes it to its destination as a fast-start to fast-start call. This can result in one-way media if:

- The destination of the hairpin call is in the same realm as the originating slow-start to fast-start call
- The realm reference in the first bullet item is configured to disable in-realm media management
- The called endpoint accepts the proposed fast-start logical channels

The enhancements to the Oracle Enterprise Session Border Controller's behavior described in this section show how the Oracle Enterprise Session Border Controller follows additional procedures when setting up a hairpin flow to avoid one-way media when media release occurs.

For H.323 calls, the Oracle Enterprise Session Border Controllerestablishes media using the H.245 procedures described in the H.245 ITU-T recommendation: control protocol for multimedia communication. It also uses the Fast Connect procedure defined in the H.323 ITU-T recommendation: packet-based multimedia communication systems.

The latter ITU-T recommendation allows a calling endpoint to send a Setup message that contains a fastStart element, a sequence of OLC structures that describe the calling endpoint's proposed forward/reverse logical channels. If the called endpoint accepts this proposal, then logical channels are established.

When the Oracle Enterprise Session Border Controller translates a call originating in slow-start to fast-start, it uses a Fast Connect procedure in the outgoing leg by sending an outgoing Setup that includes a fastStart element with one or more OLC structures. But when the Oracle Enterprise Session Border Controller constructs this message, it is

unaware of whether the call will become hairpinned or if media release will occur. Because it does not yet have this information, the Oracle Enterprise Session Border Controller sets the Network Address and the TSAP identifier in the OLC structures to the ingress IP address and port of a corresponding media flow allocated for media traveling between the calling and called endpoints. So if the called endpoint accepts the fastStart the Oracle Enterprise Session Border Controller proposes, the called endpoint would send its media to the Oracle Enterprise Session Border Controller. After acceptance, the system starts H.245 procedures on the slow-start side of the call to set up logical channels on that side. Then the Oracle Enterprise Session Border Controller updates the IP address and port of the media flows using OLC and OLCAck messages received from the calling endpoint.

This procedure works well for endpoints that are not in the same realm, or that are in the same realm for which media management is disabled, because each endpoint must send its media through the Oracle Enterprise Session Border Controller. When the endpoints are in the same realm and when media management is enabled, however, the Oracle Enterprise Session Border Controller must perform additional steps for media release in slow-start to fast-start calls.

To support media release in slow-start to fast-start calls, the Oracle Enterprise Session Border Controller performs a hold-and-resume procedure on the fast-start side. After it establishes channels on the slow-start side and if it detects media release being enabled, the Oracle Enterprise Session Border Controller sends and empty TCS to the fast-start side to put that side on hold. Then the called endpoint closes all the logical channels it previously opened in the Fast Connect procedure and stops transmitting to them. And the Oracle Enterprise Session Border Controller also closes it logical channels. Once the channels are closed, the Oracle Enterprise Session Border Controller resumes the call by sending a new, restricted TCS to the fast-start side. The restricted TCS only contains the receive and transmit capabilities of the codecs types that the called endpoint accepted in the Fast Connect procedure, and it forces the called endpoint to re-open logical channels of the same codec types accepted in the Fast Connect procedure. Once it receives and OLC from the called endpoint, the Oracle Enterprise Session Border Controller sends on OLCAck with the Network Address and TSAP identifier for the logical channel from the calling endpoint. Then the Oracle Enterprise Session Border Controller re-opens logical channels (of the same codec types that it open in the Fast Connect procedure). If the called endpoint has not changed its Network Address and TSAP identifier for its logical channels, media is re-established after the Oracle Enterprise Session Border Controller and the called endpoint exit the hold state. The last steps is for the Oracle Enterprise Session Border Controller to re-sends the full TCS message from the calling to the called endpoint to inform the called endpoint of the full capabilities of the calling endpoint.

Dependencies

This feature depends on the following assumptions:

- The H.323 endpoint supports the third-party-initiated pause and re-routing feature.
- The H.323 endpoint does not change its Network Address and TSAP identifier when it re-opens the logical channels.
- The H.323 endpoint does not immediately tear down the call when there is not established logical channel in the call.

Hold-and-Resume Procedure

The hold-and-resume procedure has three states:

- Media Hold—Starts when the Oracle Enterprise Session Border Controller sends the empty TCS to the called endpoint to put it on hold.

  When it detects media release, the Oracle Enterprise Session Border Controller puts the called endpoint on hold. It can only do so if it has exchanged the TCS/TCSAck messages and completed master-slave determination with the calling endpoint.

  When the Oracle Enterprise Session Border Controllerreceives a TCSAck in response to the empty TCS that it sent to the called endpoint, it closes the logical channels it opened as part of the Fast Connect procedure; the called endpoint likewise closes its logical channels. The two then exchange CLC and CLCAck messages, which signals the start of the Media Resume state.
- Media Resume—Starts when the Oracle Enterprise Session Border Controllersends a restricted TCS to resume the call.

  The restricted TCS the Oracle Enterprise Session Border Controller sends contains only the receive/transmit capabilities of the codec types previously accepted by the called endpoint in the Fast Connect procedure. This

forces the called endpoint to re-open logical channels of the same codec type that were previously accepted in the Fast Connect procedure.

After sending this TCS, the system is ready (as specified in the ITU-T recommendations) to take part on the master-slave determination (MSD) process. However, the called party and not the Oracle Enterprise Session Border Controller initiates the MSD if it is required. The MSD is completed if necessary. Alternately, the called endpoint can start to re-open its logical channels. When it receives the first OLC from the called endpoint, the Oracle Enterprise Session Border Controller also starts to re-open its logical channels.

- Media Complete—Starts when all the logical channels that the Oracle Enterprise Session Border Controller re-opens are acknowledged by the called endpoint.

When it enters the Media Complete state, the Oracle Enterprise Session Border Controller updates the called endpoint with the full capabilities of the calling endpoint by sending the full TCS.

H.323 and IWF Call Forwarding

This section describes the Oracle Enterprise Session Border Controller's H.323 and IWF Call Forwarding feature, which is supported for H.323 calls and for calls initiated in SIP that require interworking to H.323.

Previous Behavior

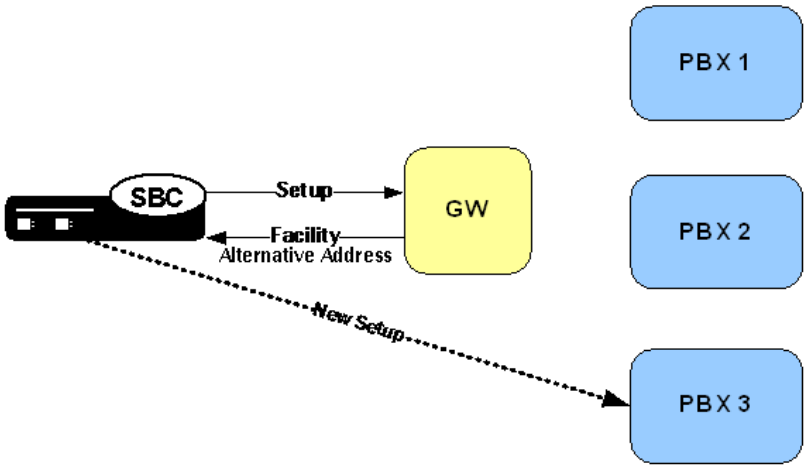
Prior to Release 4.1, the Oracle Enterprise Session Border Controller did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

- Fail to release the initial call and initiate the forwarded call
- Drop the entire call when the remote endpoint for the call tore down the session

New Behavior

In the diagram below, you can see that the Oracle Enterprise Session Border Controller sends the initial Setup message to the gateway, and the gateway returns the Facility message with an alternate address for forwarding. Rather than engaging in its former behavior, the Oracle Enterprise Session Border Controller now releases the call with the gateway and sends a new Setup to the alternate address from the Facility message.

This new Setup up has no effect on the first call leg, which remains connected.



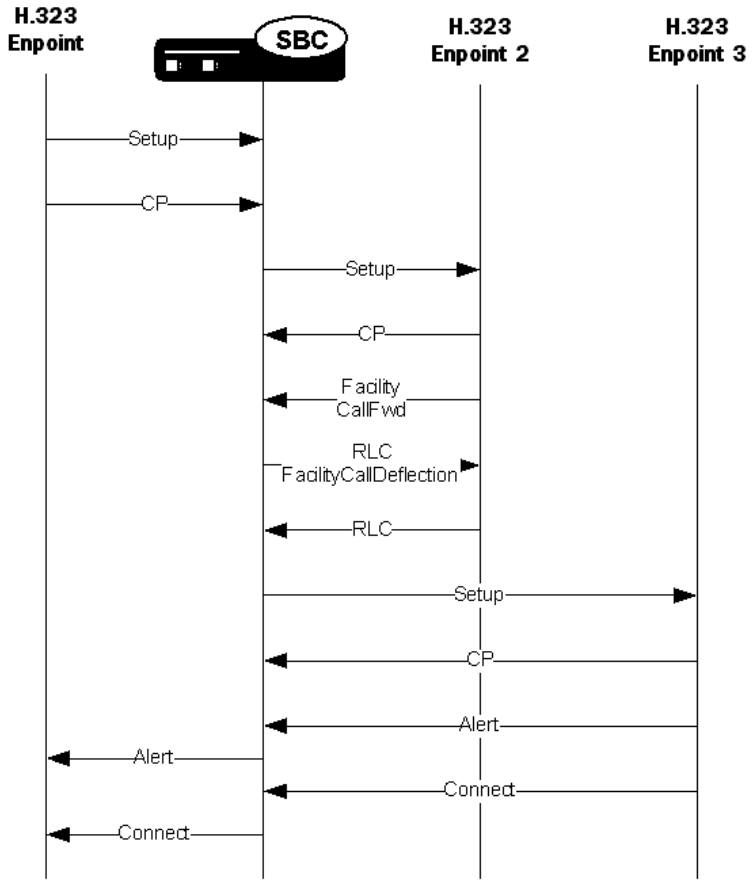
When it receives a Facility message with the reason CallForwarded, the Oracle Enterprise Session Border Controller looks for an alternate transport address in the Facility's alternativeAddress or alternativeAliasAddress element. The Oracle Enterprise Session Border Controller releases the egress call with the reason facilityCallDeflection. Then it takes one of two courses of action:

- If it does not find an alternative address, the Oracle Enterprise Session Border Controller releases the ingress call (with the reason facilityCallDeflection).

- If it finds an alternative address and the egress call has not been alerted or answered, the Oracle Enterprise Session Border Controller at this point tries to initiate a new egress call. The Oracle Enterprise Session Border Controller uses the alternative alias address to populate the calledPartyNumber information element (IE) and the destination address of the new Setup.

H.323 Sample Call Flow

The following diagram shows how the H.323 Call Forwarding feature works in a purely H.323 environment.



H.323 NOTIFY Support

To inform another call party of a certain event or communicate information to it, and H.323 entity might send a NOTIFY message. For example, a gateway might send a NOTIFY message to inform the calling party of a display name for a transferee. in previous releases, the Oracle Enterprise Session Border Controller did not process such a NOTIFY message, blocking the message from reaching its intended destination.

The Oracle Enterprise Session Border Controller supports the NOTIFY message so that it can pass through and reach its intended destination.

Caveats

The Oracle Enterprise Session Border Controller does not support interworking the NOTIFY message to a SIP message for calls that require interworking between H.323 and SIP; this support is for pure H.323 calls only.

H.323 H.239 Support for Video+Content

The Oracle Enterprise Session Border Controller supports multiple media streams for the same payload, generic capabilities, and H.239 generic messages. As a result, these additions broaden the Oracle Enterprise Session Border Controller's support for videoconferencing, and free you from have to configure media profiles for H.323 support.

Note: These additions are supported for H.323-H.323 traffic only. These additions do not support SIP-H.323 interworking (IWF), so you still need to configure media profiles for that application.

Multiple Media Streams with the Same Payload

In releases prior to S-C6.2.0, the Oracle Enterprise Session Border Controller supports multiple audio-video-data streams only if those streams use different payload types. The Oracle Enterprise Session Border Controller's behavior is extended to provide this support as of Release S-C6.2.0. The Oracle Enterprise Session Border Controller identifies extendedVideoCapability used to establish an additional channel for H.239-compliant endpoints, an OLC that was formerly not supported.

Support for Generic Capabilities

This feature identifies the OIDs shown in the table below and uses the dynamicPayload type to from the incoming OLC to generate its own OLC. So you no longer meed media profiles for: genericAudio, genericVideo, and genericData.

Capability Name	Capability Class	Capability Identifier
H.283	Data protocol	{itu-t (0) recommendation (0) h (8) 283 generic-capabilities (1) 0}
G.722.1	Audio protocol	{itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) 0}
G.722.1 Extension	Audio protocol	{itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) extension (1) 0}
H.324	Data protocol	{itu-t (0) recommendation (0) h (8) 324 generic-capabilities (1) http (0)}
H.263	Video protocol	{itu-t (0) recommendation (0) h (8) 263 generic-capabilities (1) 0} Note: Use of this capability to signal H.263 "Profiles and Levels" per Annex X/H.263 should always be accompanied in parallel by the signalling of the same modes in H263VideoCapability. This is necessary to ensure that systems which do not recognize the H.263 generic capabilities continue to interwork with newer systems.
H.224	Data protocol	{itu-t (0) recommendation (0) h (8) 224 generic-capabilities (1) 0}
G.722.2	Audio protocol	{itu-t (0) recommendation (0) g (7) 7222 generic-capabilities (1) 0}
G.726	Audio protocol	{itu-t (0) recommendation (0) g (7) 726 generic-capabilities (1) version2003 (0)}
H.241/H.264	Video protocol	{itu-t (0) recommendation (0) h (8) 241 specificVideoCodecCapabilities (0) h264 (0) generic-capabilities (1)}
H.241/H.264	Video protocol	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) RFC3984NonInterleaved(1)}

Capability Name	Capability Class	Capability Identifier
H.241/H.264	Video protocol	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) RFC3984Interleaved(2)}

Support for H.239 Generic Messages

This section describes the Oracle Enterprise Session Border Controller's support for H.239 Generic Messages.

Generic Message	Description
Generic Request	• flowControlReleaseRequest—Used when a device wants to add a channel toward an MCU that has sent multipointConference, or if the device wants to increase a channel bit rate when the channel is flow-controlled. The message has the channelId, which is the logicalChannelNumber of the channel. The Oracle Enterprise Session Border Controller proxies this message, replacing the channelId with the logicalChannelNumber of its channel. • presentationTokenRequest—Request by the sender to acquire the indicated token. The message has the channelId, which is the logicalChannelNumber of the channel. The Oracle Enterprise Session Border Controller proxies this message, replacing the channelId with the logicalChannelNumber of its channel.
Generic Response	• flowControlReleaseResponse—Sent in response to the flowControlReleaseRequest, either acknowledging or rejecting the request. The "acknowledge" response indicates the far-end device intends to make a best-effort attempt to comply with the request. The exact bit rate requested may not be allocated. The reject response indicates that the far-end device does not intend to comply with the request. The response contains the channelId that was sent in the request. While proxying the response, the Oracle Enterprise Session Border Controller will replace the channelId with the channelId it received in the request. • presentationTokenResponse—Sent in response to the presentationTokenRequest. The response will either confirm or reject the assignment of the indicated token to the sender of the presentationTokenRequest. The response contains the channelId that was received in the request. While proxying the response, the Oracle Enterprise Session Border Controller will replace the channelId with the channelId it received in the request.
Generic Command	• presentationTokenRelease—Sent by the device holding the token in order to relinquish the token. The message has the channelId, which is the logicalChannelNumber of the channel. The Oracle Enterprise Session Border Controller proxies this message, replacing the channelId with the logicalChannelNumber of its channel.
Generic Indication	• presentationTokenIndicateOwner—Indicates who owns the token. The message has the channelId, which is the logicalChannelNumber of the channel. The Oracle Enterprise Session Border Controller proxies this message, replacing the channelId with the logicalChannelNumber of its channel.

Support for Miscellaneous Indication

An endpoint sends a miscellaneous indication to send (logicalChannelActive) or stop (logicalChannelInactive) live video streams. The message has a channelId, which is the channel's logicalChannelNumber. The Oracle Enterprise Session Border Controller proxies this message, replacing the channelId with the logicalChannelNumber of its own channel.

SIP-H.323 interworking with Dynamic Payload Types

The SIP and H.323 Protocols use Internet multimedia signaling over IP, and both use the Real-Time Transport Protocol (RTP) for transferring realtime audio/video data. The interworking function (IWF) provides a means of converting translation and signaling protocols and session descriptions between SIP and H.323. However, SIP and H.323 provide different mechanisms when exchanging payload types for media during IWF calls. Therefore, the International Telecomunications Union (ITU) modified the ITU H.245 recommendations in H.245 v16 to include a new "Dynamic Payload Type Replacement" capability that resolves this payload type conflict. This new capability provides a way for an H.323 endpoint to specify the payload type of a media stream for which the endpoint is willing to receive through the OLCacknowledgment (OLC-ACK) message in an audio/video call flow.

The Oracle Enterprise Session Border Controller supports this new "Dynamic Payload Type Replacement" capability by ensuring interworking of SIP and H.323 when audio/video call flows use dynamic payload types. The Oracle Enterprise Session Border Controller checks for the presence of this capability in the incoming TCS request. If it finds this capability in the TCS request, it sends an Open Logic Channel Acknowledgement (OLC-ACK) response with the payload type it is willing to receive.

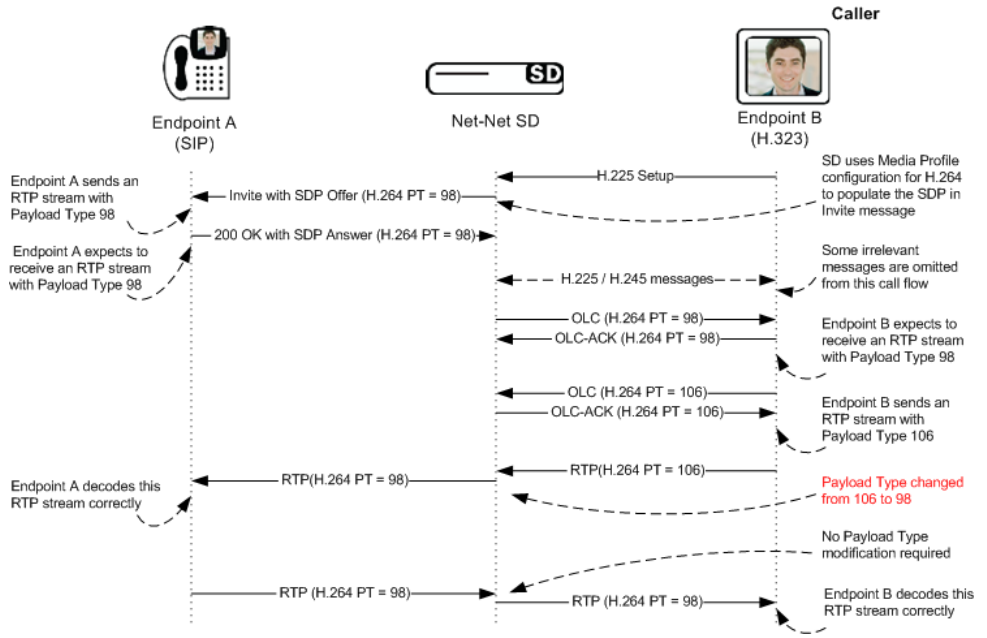
Note: The Oracle Enterprise Session Border Controller always returns an OLC-ACK with a dynamic payload type value that it received in the incoming Session Description Protocol (SDP) from the SIP endpoint.

For devices that don't support the H.245 v16 recommendations, the Terminal Capability Set (TCS) request from the H.323 endpoint does not have the "Dynamic Payload Type Replacement" capability present. Therefore, the Oracle Enterprise Session Border Controller rewrites the payload type within the RTP packets when these packets traverse the Oracle Enterprise Session Border Controller. When devices in a session negotiate different payload types between SIP and H.323 packets, the RTP streams that they receive, always have the expected payload type in the RTP header.

Note: The Oracle Enterprise Session Border Controller always maps the payload type on the RTP stream received from the H.323 endpoint, and sends it to the SIP endpoint for both audio and video. The Oracle Enterprise Session Border Controller does not support mapping of payload types in audio streams with 2833 DTMF packets.

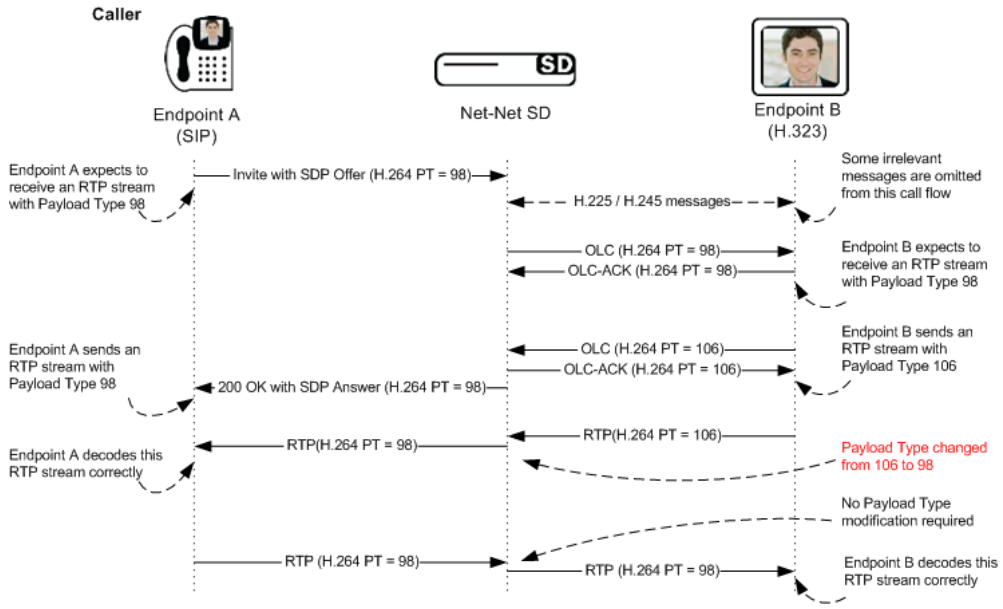
Figure 1a and 1b below shows the call flow from an H.323 Endpoint B to a SIP Endpoint A, and from a SIP Endpoint A to an H.323 Endpoint B, respectively. These illustrations show the negotiation of different dynamic payload types for the video streams but the Codec negotiated is the same. The Oracle Enterprise Session Border Controller dynamically replaces the payload type in the RTP header of the video stream received from the H.323 endpoint.

Figure 1a Endpoint B calling Endpoint A (H.323 endpoint does not have "Dynamic Payload Type Replacement" Capability)



The H.323 Endpoint B is not H.245 v16 compliant, and hence payload type replacement needs to be done in the RTP packets.

Figure 1b Endpoint A calling Endpoint B (H.323 endpoint does not have Dynamic Payload Type Replacement Capability)

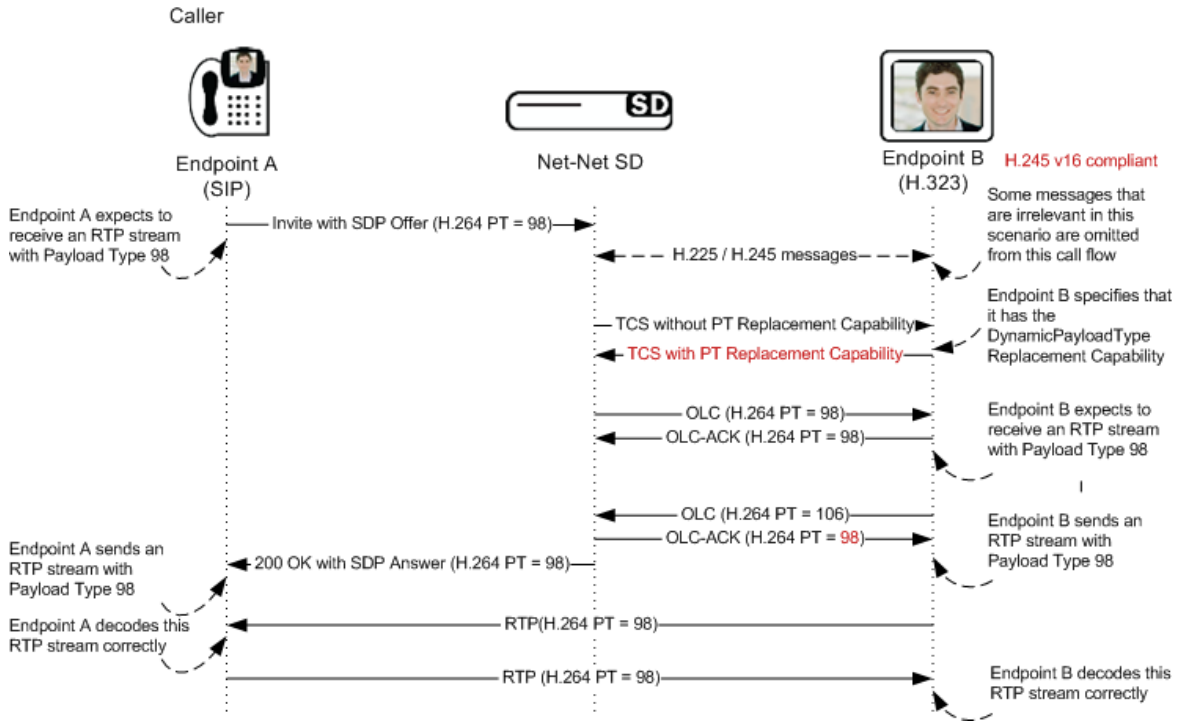


The H.323 Endpoint B is not H.245 v16 compliant, and therefore payload type replacement needs to be done in the RTP packets.

There is no concept of H.245 compliance for the SIP Endpoint A.

Figure 2a shows the call flow of SIP Endpoint A calling an H.323 Endpoint B using slow start. The Oracle Communications Session Delivery Manager modifies the dynamic payload type in the OLC-ACK based on payload type received in the incoming SDP OFFER in the "INVITE" message.

Figure 2a Endpoint A calling Endpoint B (H.323 endpoint has TCS with Dynamic Payload Type Replacement Capability)

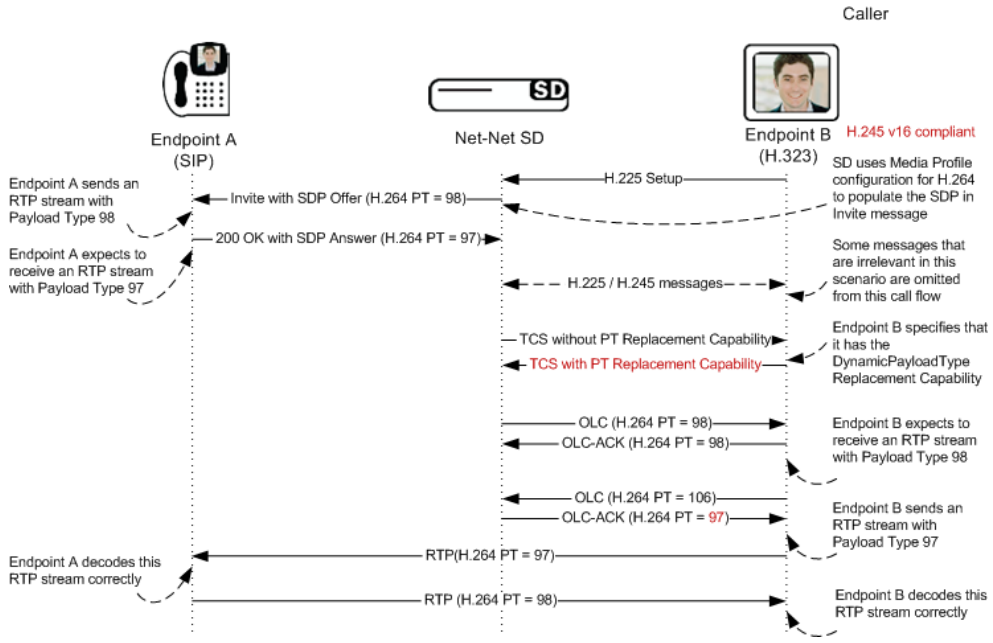


The H.323 Endpoint B is H.245 v16 compliant.

There is no concept of H.245 compliance for the SIP Endpoint A.

Figure 2b shows the call flow an H.323 Endpoint B using slow start, calling a SIP Endpoint A. The Oracle Communications Session Delivery Manager modifies the dynamic payload type in the OLC-ACK based on payload type received in the incoming SDP ANSWER in the "200 OK" message.

Figure 2b Endpoint B calling Endpoint A (H.323 endpoint here has TCS without “Dynamic Payload Type Replacement” Capability)



Video Conferencing Support for Polycom Terminals

The Oracle Enterprise Session Border Controller in a video conferencing environment with Polycom H323 terminals and a Polycom MCU (Multipoint Conferencing Unit) relays H.239/H.245. The Oracle Enterprise Session Border Controller implements the following messages appropriately:

- Miscellaneous command message with subtype such as multiPointModeCommand, cancelMultipointModeCommand
- Conference Indication message with subtype such as terminalNumberAssign, terminalYouAreSeeing

ACLI Signaling Mode Configuration Examples

The following ACLI displays provide examples of the Signaling Modes of Operation described earlier in this chapter.

Configuration Fields and Values for B2BGW Signaling

This example provides is a sample for the Back-to-Back Gateway Signaling mode of operation.

```
h323-config
        state                           enabled
        log-level                       INFO
        response-tmo                    4
        connect-tmo                     32
h323-stack
        name                            zone1
        state                           enabled
        isgateway                       enabled
        realm-id                        zone1realm
        assoc-stack                     zone2
        local-ip                        x.x.x.x (IP address of VGW-A)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               enabled
        multicast                       224.0.1.41:1718
        gatekeeper                      x.x.x.x (IP address of GkZone1)
        gk-identifier                   gk-zone1.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone2
        state                           enabled
        isgateway                       enabled
        realm-id                        DomainCrealm
        assoc-stack                     zone1
        local-ip                        x.x.x.x(IP address of VGW-C)
        max-calls                       200
```

```
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=acme01
        ras-port                        1719
        auto-gk-discovery               enabled
        multicast                       224.0.1.41:1718
        gatekeeper                      x.x.x.x(IP address of GkZONE2)
        gk-identifier                   gk-zone2.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone3
        state                           enabled
        isgateway                       enabled
        realm-id                        zone3realm
        assoc-stack                     zone4
        local-ip                        x.x.x.x(IP address of VGW-B)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               enabled
        multicast                       224.0.1.41:1718
        gatekeeper                      x.x.x.x(IP address of GkZone3)
        gk-identifier                   gk-zone3.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone4
        state                           enabled
        isgateway                       enabled
        realm-id                        DomainCrealm
        assoc-stack                     zone3
        local-ip                        x.x.x.x(IP address of VGW-D)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               enabled
```

```
        multicast                       224.0.1.41:1718
        gatekeeper                      x.x.x.x(IP address of GkZone4)
        gk-identifier                   gk-zone4.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
```

Back-to-Back Gatekeeper Proxy and Gateway

This example provides is a sample for the Back-to-Back Gateway Proxy and Gateway mode of operation.

```
h323-config
        state                           enabled
        log-level                       INFO
        response-tmo                    4
        connect-tmo                     32
h323-stack
        name                            zone1
        state                           enabled
        isgateway                       disabled
        realm-id                        zone1realm
        assoc-stack                     zone2
        local-ip                        x.x.x.x(IP address of VGW-A/GK-A)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               disabled
        multicast                       0.0.0.0:0
        gatekeeper                      x.x.x.x(IP address of GkZone1)
        gk-identifier                   gk-zone1.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone2
        state                           enabled
        isgateway                       disabled
        realm-id                        DomainCrealm
        assoc-stack                     zone1
        local-ip                        x.x.x.x(IP address of VGW-C/GK-C)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
```

```
        terminal-alias
                                  h323-ID=acme01
        ras-port                  1719
        auto-gk-discovery         disabled
        multicast                 0.0.0.0:0
        gatekeeper                x.x.x.x(IP address of GkZONE2)
        gk-identifier             gk-zone2.acme.com
        q931-port                 1720
        alternate-transport
        q931-max-calls            200
        h245-tunneling            enabled
        fs-in-first-msg           disabled
        call-start-fast           disabled
        call-start-slow           disabled
        media-profiles
        process-registration      disabled
        anonymous-connection      disabled
        proxy-mode
        filename
h323-stack
        name                      zone3
        state                     enabled
        isgateway                 disabled
        realm-id                  zone3realm
        assoc-stack               zone4
        local-ip                  x.x.x.x(IP address of VGW-B/GK-B)
        max-calls                 200
        max-channels              10
        registration-ttl          0
        terminal-alias
                                  h323-ID=private
        ras-port                  1719
        auto-gk-discovery         disabled
        multicast                 0.0.0.0:0
        gatekeeper                x.x.x.x(IP address of GkZone3)
        gk-identifier             gk-zone3.acme.com
        q931-port                 1720
        alternate-transport
        q931-max-calls            200
        h245-tunneling            enabled
        fs-in-first-msg           disabled
        call-start-fast           disabled
        call-start-slow           disabled
        media-profiles
        process-registration      disabled
        anonymous-connection      disabled
        proxy-mode
        filename
h323-stack
        name                      zone4
        state                     enabled
        isgateway                 disabled
        realm-id                  DomainCrealm
        assoc-stack               zone3
        local-ip                  x.x.x.x(IP address of VGW-D/GK-D)
        max-calls                 200
        max-channels              10
        registration-ttl          0
        terminal-alias
        h323-ID=private
        ras-port                  1719
        auto-gk-discovery         disabled
        multicast                 0.0.0.0:0
        gatekeeper                x.x.x.x(IP address of GkZone4)
```

```
        gk-identifier                   gk-zone4.acme.com
        alternate-transport
        q931-port                       1720
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
```

Interworking Gatekeeper-Gateway

This example provides is a sample for the Interworking Gatekeeper-Gateway mode of operation.

```
h323-config
        state                           enabled
        log-level                       INFO
        response-tmo                    4
        connect-tmo                     32
h323-stack
        name                            zone1
        state                           enabled
        isgateway                       disabled
        realm-id                        zone1realm
        assoc-stack                     zone2
        local-ip                        x.x.x.x(IP address of VGW-A/GK-A)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               disabled
        multicast                       0.0.0.0:0
        gatekeeper                      x.x.x.x(IP address of GkZone1)
        gk-identifier                   gk-zone1.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone2
        state                           enabled
        isgateway                       enabled
        realm-id                        DomainCrealm
        assoc-stack                     zone1
        local-ip                        x.x.x.x(IP address of VGW-C)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=acme01
```

```
        ras-port                        1719
        auto-gk-discovery               enabled
        multicast                       0.0.0.0:0
        gatekeeper                      0.0.0.0:0
        gk-identifier                   gk-zone2.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone3
        state                           enabled
        isgateway                       disabled
        realm-id                        zone3realm
        assoc-stack                     zone4
        local-ip                        x.x.x.x(IP address of VGW-B/GK-B)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               disabled
        multicast                       0.0.0.0:0
        gatekeeper                      x.x.x.x(IP address of GkZone3)
        gk-identifier                   gk-zone3.acme.com
        q931-port                       1720
        alternate-transport
        q931-max-calls                  200
        h245-tunneling                  enabled
        fs-in-first-msg                 disabled
        call-start-fast                 disabled
        call-start-slow                 disabled
        media-profiles
        process-registration            disabled
        anonymous-connection            disabled
        proxy-mode
        filename
h323-stack
        name                            zone4
        state                           enabled
        isgateway                       enabled
        realm-id                        DomainCrealm
        assoc-stack                     zone3
        local-ip                        x.x.x.x(IP address of VGW-D)
        max-calls                       200
        max-channels                    10
        registration-ttl                0
        terminal-alias
                                        h323-ID=private
        ras-port                        1719
        auto-gk-discovery               disabled
        multicast                       0.0.0.0:0
        gatekeeper                      x.x.x.x(IP address of GkZone4)
        gk-identifier                   gk-zone4.acme.com
```

Additional Information

This section contains detailed tables to use as a reference when you are learning about H.323 features or when you are configuring them.

About Payload Types

You set the payload type when you are configuring a media profile to support Slow Start to Fast Start Translation.

When you configure media profiles, you might need set the payload type to identify the format in the SDP m lines. For RTP/AVP, the default transport method of a media profile configuration, this will be the RTP payload type number. Newer codecs have dynamic payload types, which means that they do not have an assigned payload type number.

When you use RTP/AVP as the transport method, you should only set the payload type when there is a standard payload type number for the encoding name; otherwise, leave the payload type blank.

The Oracle Enterprise Session Border Controller uses the payload type value to determine the encoding type when SDP identifies the standard payload type in the m line, but does not include an a=rtpmap entry. These are two equivalent SDPs:

c=IN IP4 192.0.2.4

```
m=audio 0 RTP/AVP 0
```

c=IN IP4 192.0.2.4

```
m=audio 0 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

The first does not include the RTP map entry, but uses the standard payload type of 0. If the Oracle Enterprise Session Border Controller receives an SDP like the first, it uses the payload type 0 to locate the corresponding media profiles configuration. When an a=rtpmap is present, the Oracle Enterprise Session Border Controller uses the encoding name in the a=rtpmap line to find the media profile configuration and does not consider the payload type number.

Payload Types for Standard Audio and Visual Encodings

The following is a table of standard audio and visual payload encodings defined in H. Schulzrinne, GND Fokus, RTP Profile for Audio and Visual Conferences with Minimal Control, RFC 1890, and in the *RTP Parameters* document in IANA's Directory of Generally Assigned Numbers.

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
0	PCMU	A	8000
1	1016	A	8000
2	G721	A	8000
3	GSM	A	8000
4	G723	A	8000
5	DVI4	A	8000
6	DVI4	A	16000
7	LPC	A	8000
8	PCMA	A	8000
9	G722	A	8000
10	L16	A	44100

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
11	L16	A	44100
12	QCELP	A	8000
13	reserved	A	
14	MPA	A	90000
15	G728	A	8000
16	DVI4	A	11025
17	DVI4	A	22050
18	G729	A	8000
19	reserved	A	
20	unassigned	A	
21	unassigned	A	
22	unassigned	A	
23	unassigned	A	
dyn	GSM-HR	A	8000
dyn	GSM-EFR	A	8000
dyn	L8	A	var.
dyn	RED	A	
dyn	VDVI	A	var.
24	unassigned	V	
25	CelB	V	90000
26	JPEG	V	90000
27	unassigned	V	
28	nv	V	90000
29	unassigned	V	
30	unassigned	V	
31	H261	V	90000
32	MPV	V	90000
33	MP2T	AV	90000
34	H263	V	90000
35-71	unassigned	?	
72-76	reserved for RTCP conflict avoidance	N/A	N/A
77-95	unassigned	?	
96-127	dynamic	?	
dyn	BT656	V	90000

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
dyn	H263-1998	V	90000
dyn	MP1S	V	90000
dyn	MP2P	V	90000
dyn	BMPEG	V	90000

About RAS Message Treatment

When you enabled the H.323 Registration Proxy, the Oracle Enterprise Session Border Controller modifies and deletes certain fields as outlines in the table below. The Oracle Enterprise Session Border Controller sends on any fields that are not listed in this table without modifying or deleting them.

Note: Although the Oracle Enterprise Session Border Controller forwards a field, it does not always support the feature related to that field.

Field Name	Message	Deleted	Modified	Value Used in Modification
alternateEndpoints	RRQ, URQ, ACF	X		
alternateGatekeeper	RCF, URQ	X		
altGKInfo	RRJ, URJ, DRJ	X		
alternateTransportAddresses	RRQ, ARQ, ACF	X		
callModel	ARQ		X	direct
	ACF		X	gatekeeperRouted
callSignalAddress	RRQ		X	Mapped virtual CSA allocated by the system for registering the endpoint.
	RCF, ARJ		X	CSA of gatekeeper stack
	URQ		X	If URQ is from an endpoint, endpoint's mapped virtual CSA. If URQ is from a gatekeeper, real CSA of endpoint.
destCallSignalAddress	ARQ, ACF	X		
destinationInfo.transportID	ARQ, ACF	X		
destExtraCallInfo.trasnportID	ARQ, ACF	X		
discoveryComplete	RRQ		X	TRUE
endpointAlias.trasnportID	URQ	X		
endpointAliasPattern.Wwildcard.transportID	URQ			
featureServerAlias.trasnportID	RCF	X		

Field Name	Message	Deleted	Modified	Value Used in Modification
gatekeeperIdentifier	RRQ		X	Gatekeeper identifier of the gateway stack, either configured in the H.323 gateway stack or discovered dynamically.
maintainConnection	RRQ, RCF		X	FALSE
mutlipleCall	RRQ, RCF		X	FALSE
preGrantedARQ.alternateTransportAddresses	RCF	X		
preGrantedARQ.useSpecifiedTransport	RCF	X		
rasAddress	RRQ		X	Mapped virtual RAS address allocated by the system for registering endpoint
remoteExtentsionAddress.trasnportID	ARQ, ACF	X		
srcCallSignalAddress	ARQ	X		
srcInfo.transportID	ARQ	X		
supportedH248Packages	RRQ	X		
supportsAltGK	RRQ	X		
supportedPrefixes.prefic.transportID	RCF, URQ	X		
terminalAlias.transportID	RRQ	X		
terminalAliasPattern.wilcard.transportID	RRQ	X		
willRespondToIIRR	RCF, ACF	X		
willSupplyUUIEs	RRQ, ARQ			
uuiesRequested	ACF		X	FALSE
setup			X	FALSE
callProceeding			X	FALSE
connect			X	FALSE
alerting			X	FALSE
information			X	FALSE
releaseComplete			X	FALSE
facility			X	FALSE
progress			X	FALSE
empty			X	FALSE
...,			X	FALSE
status			X	FALSE

Field Name	Message	Deleted	Modified	Value Used in Modification
statusInquiry setupAcknowledge notify			X	FALSE

11

Application Layer Gateway Services

DNS ALG

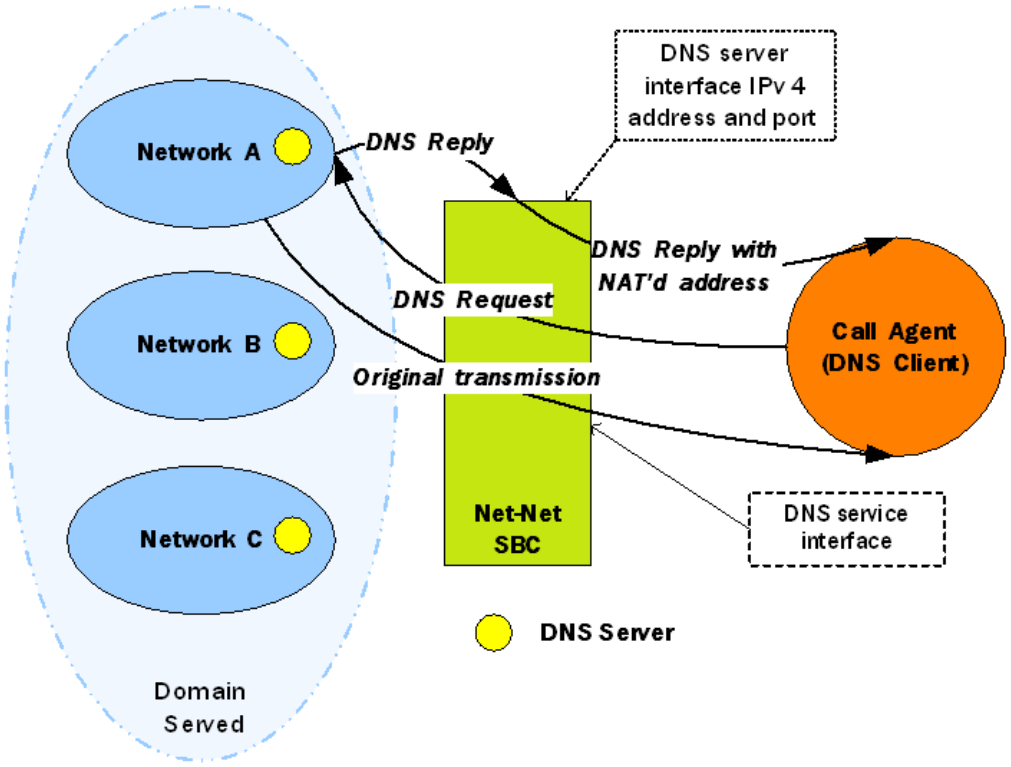
The Oracle Enterprise Session Border Controller's DNS Application Layer Gateway (ALG) feature provides an application layer gateway for DNS transactions on the Oracle Enterprise Session Border Controller. With DNS ALG service configured, the Oracle Enterprise Session Border Controller can support the appearance of multiple DNS servers on one side and a single DNS client on the other.

Overview

DNS ALG service provides an application layer gateway for use with DNS clients. DNS ALG service allows a client to access multiple DNS servers in different networks and provides routing to/from those servers. It also supports flexible address translation of the DNS query/response packets. These functions allow the DNS client to query many different domains from a single DNS server instance on the client side of the network.

The Oracle Enterprise Session Border Controller's DNS ALG service is commonly used when a DNS client (such as a call agent) needs to authenticate users. In this case, the DNS client that received a message from a certain network would need to authenticate the endpoint in a remote network. Since the DNS client and the sender of the message are on different networks, the Oracle Enterprise Session Border Controller acts as an intermediary by interoperating with both.

In the following diagram, the DNS client has received a message from an endpoint in Network A. Since the DNS client is in a different realm, however, the DNS client receives the message after the Oracle Enterprise Session Border Controller has performed address translation. Then the DNS client initiates a DNS query on the translated address. The Oracle Enterprise Session Border Controller forwards the DNS request to the DNS server in Network A, using the domain suffix to find the appropriate server. Network A's DNS server returns a response containing its IPv4 address, and then the Oracle Enterprise Session Border Controller takes that reply and performs a NAT on the private address. The private address is turned into a public one that the DNS client can use to authenticate the endpoint.



Configuring DNS ALG Service

This section tells you how to access and set the values you need depending on the configuration mechanism you choose. It also provides sample configurations for your reference.

Configuring DNS ALG service requires that you carry out two main procedures:

- Setting the name, realm, and DNS service IP interfaces
- Setting the appropriate parameters for DNS servers to use in other realms

Before You Configure

Before you begin to configure DNS ALG service on the Oracle Enterprise Session Border Controller, complete the following steps.

1. Configure the client realm that you are going to use in the main DNS ALG profile and note its name to use in this chapter's configuration process.
2. Configure the server realm that contains the DNS servers and note its name to use in this chapter's configuration process.
3. Determine the domain suffixes for the network where the DNS servers are located so that you can enter them in the domain suffix parameter.
4. Devise the NAT scheme that you want to use when the DNS reply transits the Oracle Enterprise Session Border Controller.

DNS ALG Service Name Configuration

This section explains how to configure the name of the DNS ALG service you are configuring and set its realm.

To add DNS ALG service:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type dns-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# dns-config
ACMEPACKET(dns-config)#
```

From this point, you can configure DNS ALG parameters and access this configuration's DNS server subelement. To view all DNS ALG service parameters and the DNS server subelement, enter a ? at the system prompt.

```
dns-config
        client-realm
        description                     dns-alg1
        client-address-list
        last-modified-date              2005-02-15 10:50:07
        server-dns-attributes
                server-realm
                domain-suffix
                server-address-list
                source-address
                source-port                     53
                transaction-timeout             10
                        address-translation
                                server-prefix                   10.3.0.0/16
                                client-prefix
192.168.0.0/16
```

Identity Realm and Interface Addresses

To configure the identity, realm, and IPv4 interface addresses for your DNS ALG profile:

1. description—Set a name for the DNS ALG profile using any combination of characters entered without spaces. You can also enter any combination with spaces if you enclose the whole value in quotation marks. For example: DNS ALG service.
2. client-realm—Enter the name of the realm from which DNS queries are received. If you do not set this parameter, the DNS ALG service will not work.
3. client-address-list—Configure a list of one or more addresses for the DNS server interface. These are the addresses on the Oracle Enterprise Session Border Controller to which DNS clients send queries.

   To enter one address in this list, type client-address-list at the system prompt, a Space, the IPv4 address, and then press Enter

```
ACMEPACKET(dns-config)# client-address-list 192.168.0.2
```

   To enter more than one address in this list, type client-address-list at the system prompt, and a Space. Then type an open parenthesis (( ), each IPv4 address you want to use separated by a Space, and closed parenthesis ( )), and then press Enter.

```
ACMEPACKET(dns-config)# client-address-list (192.168.0.2 196.168.1.1
192.168.1.2)
```

DNS Server Attributes

To configure attributes for the DNS servers that you want to use in the DNS ALG profile:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type dns-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# dns-config
```

4. Type server-dns-attributes and then press Enter.

```
ACMEPACKET(dns-config)# server-dns-attributes
```

From this point, you can configure DNS server parameters. To see all parameters for the DNS server, enter a ? at the system prompt.

5. server-realm—Enter the name of the realm in which the DNS server is located. This value is the name of a configured realm.
6. domain-suffix—Enter a list of one or more domain suffixes to indicate the domains you want to serve. These values are matched when a request is sent to a specific DNS server. If you leave this list empty (default), then your configuration will not work.

Note: If you want to use a wildcard value, you can start your entry to an asterisk (*) (e.g. *.com). You can also start this value with a dot (e.g., .com).

To enter one address in this list, type client-address-list at the system prompt, a Space, the domain suffix, and then press Enter

```
ACMEPACKET(server-dns-attributes)# domain-suffix acmepacket.com
```

To enter more than one address in this list, type domain-suffix at the system prompt, and a Space. Then type an open parenthesis (( ), each IPv4 address you want to use separated by a Space, and closed parenthesis ( )), and then press Enter.

```
ACMEPACKET(server-dns-attributes)# domain-suffix (acmepacket.com
acmepacket1.com acmepacket2.com)
```

7. server-address-list—Enter a list of one or more DNS IPv4 addresses for DNS servers. These DNS servers can be used for the domains you specified in the domain suffix parameter. Each domain can have several DNS servers associated with it, and so you can populate this list with multiple IPv4 addresses. If you leave this list empty (default), your configuration will not work.
8. source-address—Enter the IPv4 address for the DNS client interface on the Oracle Enterprise Session Border Controller. If you leave this parameter empty (default), your configuration will not work.
9. source-port—Enter the number of the port for the DNS client interface on the Oracle Enterprise Session Border Controller. The default value is 53. The valid range is:
   - Minimum—1025
   - Maximum—65535
10. transaction-timeout—Enter the time in seconds that the ALG should keep information to map a DNS server response back to the appropriate client request. After the transaction times out, further response to the original request will be discarded. The default value is 10. The valid range is:
    - Minimum—0
    - Maximum—999999999
11. address-translation—Enter a list of address translations that define the NAT function for the DNS servers.

    You can access the NAT parameters for the DNS servers by typing address-translation and pressing enter within the DNS server attributes configuration.

```
ACMEPACKET(dns-config)# server-dns-attributes
ACMEPACKET(server-dns-attributes)# address-translation
```

To configure the NAT, enter two values:

- server-prefix: address/prefix that will be returned by the DNS server
- client-prefix: address/prefix that to which a response is returned

Each of these is a two-part value:

- IPv4 address
- Number of bits indicating how much of the IPv4 address to match

If you do not specify the number of bits, then all 32 bits of the IPv4 address will be used for matching. If you set the number of bits to 0, then the address will simply be copied.

For example, if you set the server prefix to 10.3.17.2/16 and the client prefix to 192.168.0.0/16, then the Oracle Enterprise Session Border Controller will return an address of 192.168.17.2 to the DNS client.

```
ACMEPACKET(server-dns-attributes)# address-translation
ACMEPACKET(address-translation)# server-prefix 10.3.17.2/16
ACMEPACKET(address-translation)# client-prefix 192.168.0.0/16
```

DNS Transaction Timeout

To provide resiliency during DNS server failover, you can now enable a transaction timeout for DNS servers. If you have endpoints that are only capable of being configured with a single DNS server, this can allow DNS queries to be sent to the next configured server—even when contacting the Oracle Enterprise Session Border Controller's DNS ALG on a single IP address. So when the first server in the list times out, the request is sent to the next server in the list.

The Oracle Enterprise Session Border Controller uses the transaction timeout value set in the dns-server-attributes configuration (part of the dns-config).

DNS Transaction Timeout Configuration

To enable the DNS transaction timeout:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```

2. Type media-manager and press Enter

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type media-manager and press Enter.

   ```
   ACMEPACKET(media-manager)# media-manager
   ACMEPACKET(media-manager-config)#
   ```

4. dnsalg-server-failover—Change this parameter from disabled (default) to enabled to allow DNS queries to be sent to the next configured server—even when contacting the Oracle Enterprise Session Border Controller's DNS ALG on a single IP address. So when the first server in the list times out, the request is sent to the next server in the list. The Oracle Enterprise Session Border Controller uses the transaction timeout value set in the dns-server-attributes configuration (part of the dns-config).
5. Save your work.

Dynamic ACL for the HTTP-ALG

The dynamic Access Control List (ACL) option for HTTP-Application Layer Gateway (ALG) provides Distributed Denial of Service (DDoS) attack protection for the HTTP port.

When the dynamic ACL option is enabled, the static flow for the public listening socket defined in **http-alg > public** is created with at trust level set to **untrusted**. Each listening socket creates and manages its ACL list, which allows the listening socket to keep track of the number of received and invalid messages, the number of connections per endpoint, and so on. You can configure a different setting for each **http-alg** object.

Dynamic ACL for each endpoint is triggered by Session Initialization Protocol (SIP) registration messages. Upon receiving a SIP registration message, the SIP agent creates a dynamic ACL entry for the endpoint. If the 200 OK response is received, the ACL is promoted, allowing the HTTP message to go through the security domain. If SIP registration is unsuccessful, the ACL entry is removed and HTTP ingress messages are blocked from the endpoint. The ACL entry is removed upon incomplete registration renewal or telephone disconnect.

The following example describes the criteria and associated configuration item that result in a denied or allowed connection for both low and medium control levels.

Criteria	Associated Configuration Item	Action
Exceed total number of connections for allowed	http-alg > max-incoming-conns	Connection denied
Exceed total connections per peer	http-alg > per-src-ip-mas-incoming-conns	Connection denied
ACL not promoted	Dynamically set on SIP registration	Connection denied
Exceed maximum number of packets/sec	realm-config > maximum-signal-threshold	Connection denied and peer is demoted
Exceed maximum number of error packets	Realm-config > invalid-signal-threshold	Connection denied and peer is demoted

Oracle recommends setting **realm-config > access-control-level** to medium.

If a peer is promoted to **trusted**, the system performs DDoS checks on **max number of packets/sec** and **max number of error packets** allowed.

Demotions depend on the realm's **ream-config > access-control-trust-level** setting. For more information on **realm-config** settings, see the ACLI Configuration Guide.

If you want to configure different ACL settings for SIP traffic and for HTTP-ALG traffic, you must configure a realm for each type of traffic.

Dynamic Access Control List (ACL) Settings for the HTTP Application Layer Gateway (ALG)

You can set the following parameters for the realm specified in **http-alg > public > realm-id**.

- access-control-trust-level
- invalid-signal-threshold
- maximum-signal-threshold
- untrusted-signal-threshold
- deny-period

For more information on **realm-config** settings, see the ACLI Configuration Guide.

Enable Dynamic Access Control List (ACL) for the HTTP Application Layer Gateway (ALG)

Dynamic ACL option, which provides Distributed Denial of Service (DDoS) attack protection for the HTTP port, is an option that you must enable.

Confirm that the session manager is mapped to the Oracle Enterprise Session Border Controller.

Two ACL entires are required for each registered telephone, where one entry is used for SIP traffic and one is used for HTTP-ALG traffic.

Note: Enabling dynamic access control for HTTP-ALG traffic reduces the number of available dynamic ACL entries on the session border controller, which may reduce the number of concurrent trusted endpoints that the system can support.

1. From the command line, type configure terminal, and press ENTER.
2. Type session-router, and press ENTER.
3. Type http-alg, and press ENTER.
   The system displays a list of configured HTTP-ALG objects.
4. Type the number of the HTTP-ALG object that you want to edit, and press ENTER.
   The system displays the configuration values for the selected object.
5. Type dynamic-acl enabled, and press ENTER.

6. Optional. Type max-incoming-conns <value>, and press ENTER to set the maximum number of connections per peer IP address.
7. Optional. Type per-src-ip-max-incoming-conns <value>, and press ENTER to set the maximum number of HTTP connections per peer IP address.
8. Type Done, and press ENTER to save the HTTP-ALG values.
   The system displays the HTTP-ALG configuration.
9. Exit, Save, and Activate the configuration.

12

IWF Services

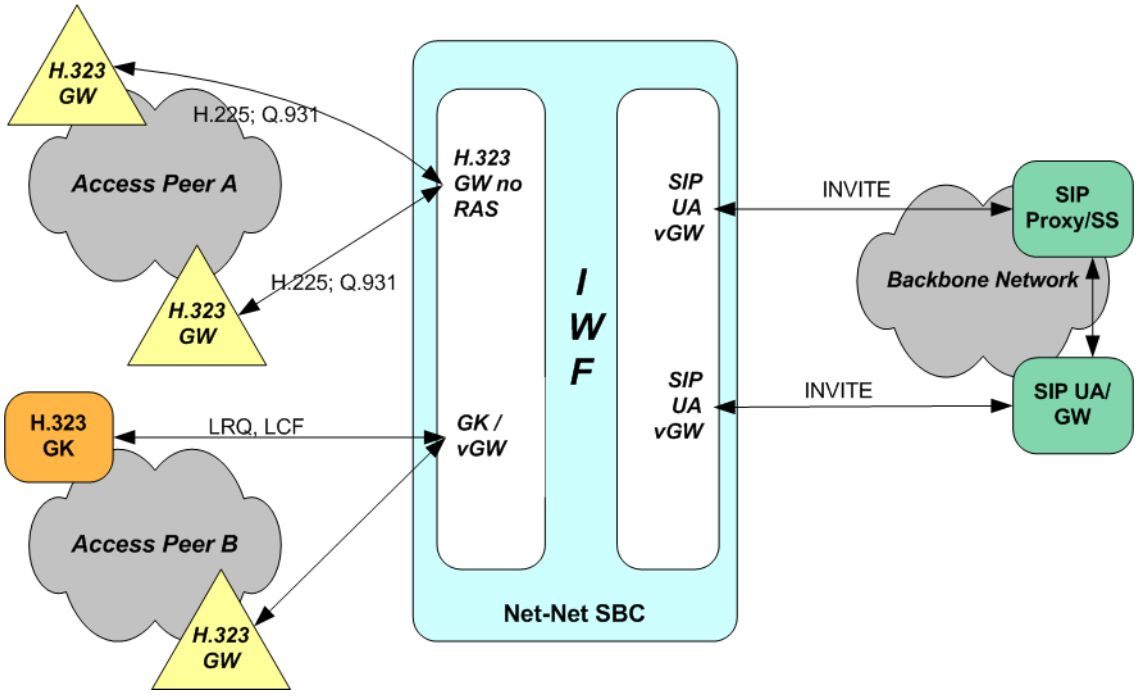
Using the Oracle Enterprise Session Border Controller's interworking function (IWF), you can interconnect SIP networks with H.323 networks. Considering the large amount of H.323 deployments already in place and the continuing emergence of SIP in new VoIP deployments, the IWF provides a much-needed solution. SIP providers can maintain a single-protocol backbone while exchanging VoIP sessions with H.323 providers.

The H.323 Signaling Services section contains information about the H.323 signaling modes of operation that the Oracle Enterprise Session Border Controller supports. The following H.323 signaling modes of operation can be used when you use the Oracle Enterprise Session Border Controller's IWF in an access or a peering solution.

- Back-to-back gateway signaling
- Interworking gatekeeper/gateway

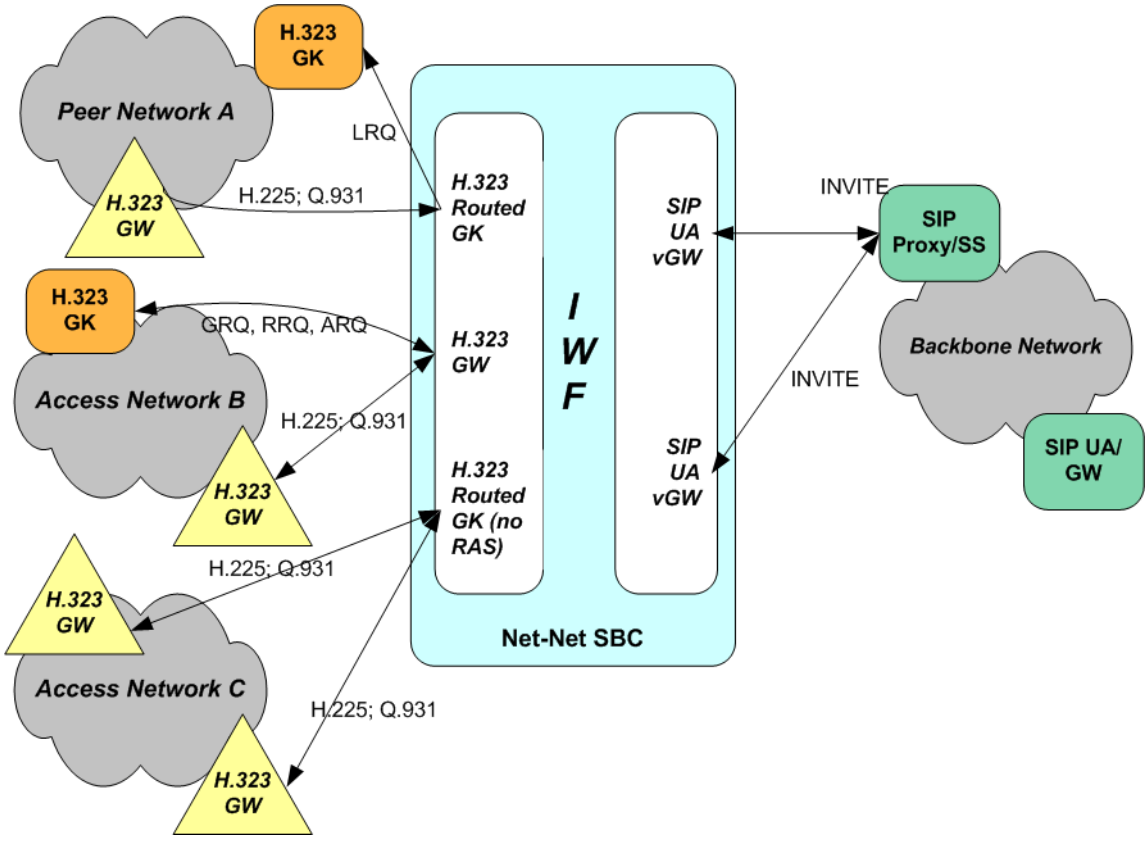
Access Network Application

You can configure your Oracle Enterprise Session Border Controller so that it provides an access solution for your network. The access solution allows SIP-based hosted communications platforms to be extended to enterprise-based H.323 systems. In the figure below, you can see different types of H.323 signaling modes being interworked with SIP. On the H.323 side, the Oracle Enterprise Session Border Controller can appear to be a gatekeeper or a gateway, depending on how you configure the H.323 interface. On the SIP side, the Oracle Enterprise Session Border Controller can appear to be a SIP UA or behave as a virtual gateway.



Networking Peering Application

In the IWF network peering solution, you can see the same network elements at work. However, the H.323 side of this IWF application shows the use of a gatekeeper controlled gateway for Peer Network B. Because this is a peering solution, the SIP side of the Oracle Enterprise Session Border Controller communicates with the SIP proxy or softswitch in the backbone network rather than with the SIP UA or SIP gateway.



SIP and H.323

The Oracle Enterprise Session Border Controller supports interworking between SIP and H.323 for H.323 Slow Start and Fast Start calls. In addition to describing IWF sessions when initiated from the H.323 side and from the SIP side (with sample call flows), this section provides information you will need when you configure SIP and H.323.

SIP H.323 Negotiation H.323 Fast Start

The Oracle Enterprise Session Border Controller can perform protocol translations for SIP and H.323 Fast Start, where media capabilities are sent with the Setup request for an H.323 session.

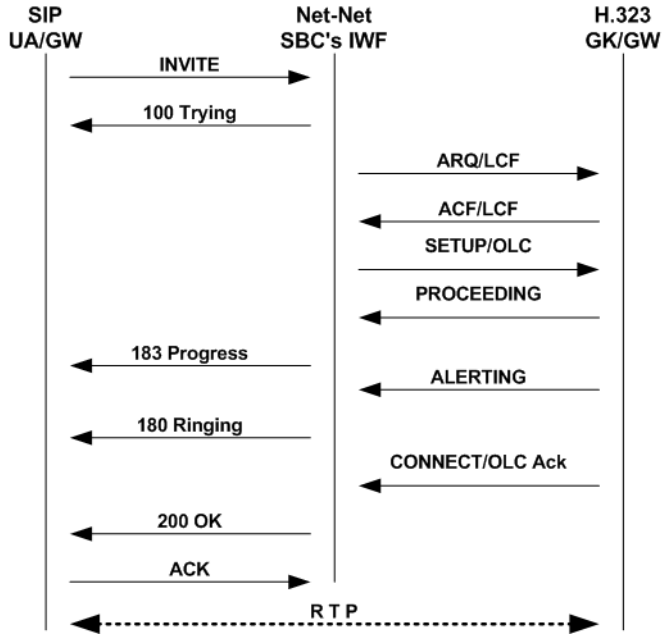
This section's call flow diagrams show how SIP and H.323 messages flow between SIP and H.323 devices, with the Oracle Enterprise Session Border Controller positioned between the two entities so it can perform translations. The following two sample scenarios with Fast Start appear in the diagrams below, although other scenarios are possible:

- Calls originating in SIP being translated to H.323 Fast Start
- Calls originating in H.323 Fast Start translated to SIP

SIP to Fast Start H.323

In the following diagram below, a SIP endpoint (such as a UA or a SIP Gateway) initiates a session by sending an INVITE message destined for an H.323 endpoint (a GK or GW). Between these entities, the system is positioned to perform interworking. The Oracle Enterprise Session Border Controller recognizes that the INVITE message is destined for an H.323 device, and returns a 100 Trying message to the SIP endpoint as it attempts to negotiate the H. 323 side of the session. This negotiation starts when the Oracle Enterprise Session Border Controller initiates the RAS process with the H.323 endpoint by sending either an ARQ or an LRQ, allowing the Oracle Enterprise Session Border Controller to determine if the H.323 endpoint will accept the session.

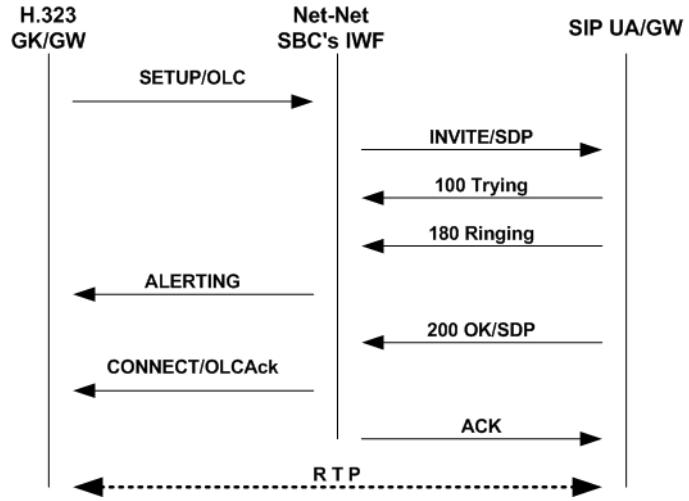
Once the H.323 endpoint responds with an ACF or LCF, the Oracle Enterprise Session Border Controller reissues the SIP INVITE on the H.323 side as an H.225 Setup, which is sent with the OLC. Then the H.323 endpoint responds with Proceeding and Alerting messages (which correspond respectively to SIP 183 Progress and 180 Ringing messages). At that point, the H.323 endpoint sends a Connect message that includes the OpenLogicalChannel message (OLC), announcing the logical channel for media flows has been set up. The Oracle Enterprise Session Border Controller converts the H.323 OLC to a SIP 200 OK. After receiving the 200 OK, the SIP endpoint sends an ACK, confirming that the session has been established. Because there is no H.323 equivalent for the SIP ACK, the Oracle Enterprise Session Border Controller does not generate a corresponding message on the H.323 side. At this point, the session is fully established and RTP flows between the endpoints.



H.323 Fast Start to SIP

In the diagram below, an H.323 endpoint (a GK or GW) initiates a session by sending a Setup request destined for a SIP endpoint (such as a UA or a SIP Gateway). Between these entities, the Oracle Enterprise Session Border Controller is positioned to perform interworking. The H.323 endpoint has completed the RAS process prior to sending the SETUP message.

The Oracle Enterprise Session Border Controller receives the Setup message and then sends a SIP INVITE on the SIP side. The SIP endpoint responds with a 100 Trying; the Oracle Enterprise Session Border Controller does not resend this message on the H.323 side. Next, the SIP endpoint issues a 180 Ringing message, which the Oracle Enterprise Session Border Controller reissues to the H.323 endpoint as an Alerting message. The SIP endpoint then sends a 200 OK, retransmitted by the Oracle Enterprise Session Border Controller as a Connect message that includes an OLC. Once the Oracle Enterprise Session Border Controller sends an ACK to the SIP endpoint, RTP flows between the endpoints.



SIP H.323 Negotiation H.323 Slow Start

The Oracle Enterprise Session Border Controller can also perform protocol translations for SIP and H.323 Slow Start, where—unlike the cases with Fast Start described above—media information is not sent with the Setup request for an H.323 session. For H.323 Slow Start, media is negotiated after the session is established.

This section's call flow diagrams show how SIP and H.323 messages flow between SIP UA/GW and an H.323 GK/GW, with the Oracle Enterprise Session Border Controller positioned between the two entities so it can perform translations. Two sample scenarios with Slow Start appear in the diagrams below:

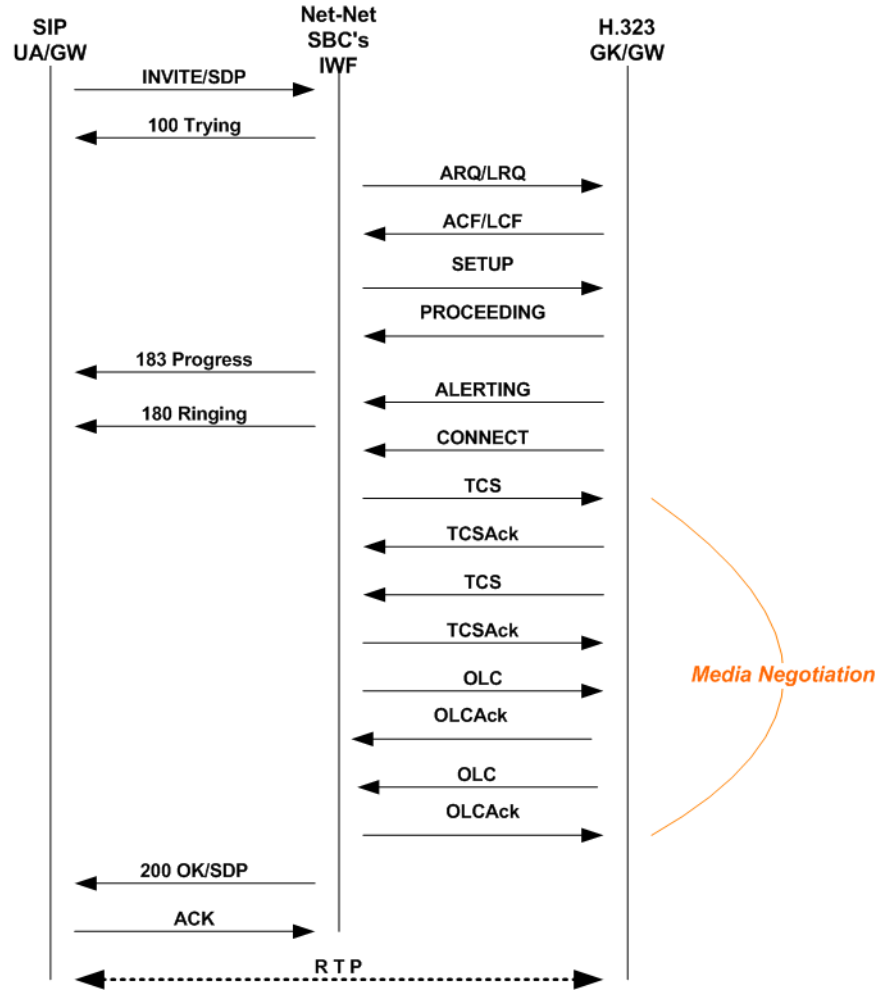
- SIP being interworked to Slow Start H.323
- Slow Start H.323 being interworked to SIP

H.323 SIP to Slow Start

In the following diagram below, a SIP endpoint (such as a UA or a SIP Gateway) initiates a session by sending an INVITE request destined for an H.323 Slow Start endpoint (a GK or GW). Between these entities, the Oracle Enterprise Session Border Controller is positioned to perform interworking.

The call flow for this type of translation works fundamentally the same way that the translation does for SIP to Fast Start H.323, with the exception of how the media is established. Media is negotiated through the exchange of TCS and OLC messages after the H.323 Connect and SIP 180 Ringing messages have been sent. The first TCS message is sent from the Oracle Enterprise Session Border Controller to the H.323 endpoint, and it contains information about media capabilities in SDP. The H.323 endpoint accepts and acknowledges this information with a TCS Ack message. Then the H.323 endpoint sends a second TCS, carrying information about the Gateway's capabilities, that the Oracle Enterprise Session Border Controller accepts and acknowledges. The H.323 endpoint and the Oracle Enterprise Session Border Controller then exchange OLC and OLC Ack messages that establish the operating mode and

Gateway capability. Finally, the Oracle Enterprise Session Border Controller completes the 200 OK/ACK sequence on the SIP side, and RTP flows between the two endpoints.

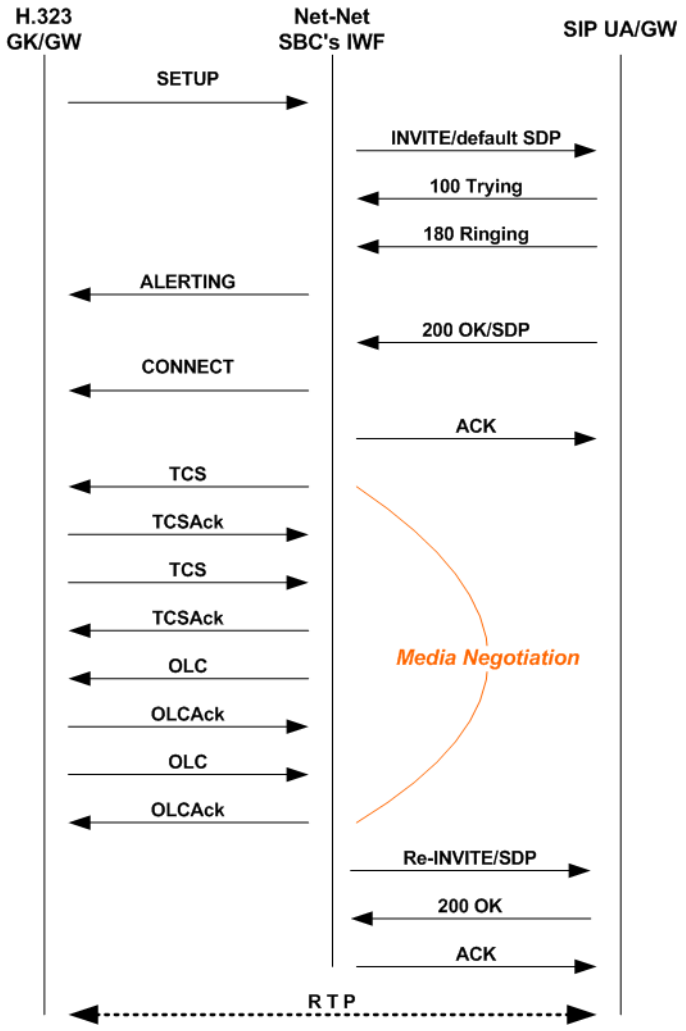


H.323 Slow Start to SIP

In the following diagram below, an H.323 endpoint (GW or GK) initiates a session by sending a Setup request destined for a SIP endpoint (such as a UA or a SIP Gateway). Between these entities, the Oracle Enterprise Session Border Controller is positioned to perform interworking. The H.323 endpoint has completed the RAS process prior to sending the SETUP message.

The call flow for this type of translation works fundamentally the same way that the translation does for H.323 Fast Start to SIP, with the exception of how the media is established. When the Oracle Enterprise Session Border Controller receives an H.323 message destined for a SIP endpoint, it sends a SIP INVITE message that includes default SDP to that SIP endpoint. The default SDP is constructed using information in the media profiles listed for the IWF configuration; if necessary, this media information is amended later in the sequence. Once the call is set up, the Oracle Enterprise Session Border Controller negotiates media with the H.323 endpoint through a series of TCS/TCS Ack and OLC/OLC Ack messages that establish the operating mode and Gateway capability.

When the Oracle Enterprise Session Border Controller completes media negotiation with the H.323 endpoint, it issues a re-INVITE to the SIP endpoint that contains the updated information needed for media transmission. In response, the SIP endpoint sends a 200 OK message that the Oracle Enterprise Session Border Controller answers with an ACK. Then RTP can flow between the two endpoints.



Status and Codec Mapping

The Oracle Enterprise Session Border Controller maps SIP and H.323 status codes as described in this section. Status and codec mapping do not require configuration; they occur transparently.

IWF Termination from H.323

When a call that requires the IWF terminates from the H.323 side, the Oracle Enterprise Session Border Controller uses the mapping scheme in the following table to determine the appropriate SIP status.

H.323 Disconnect Reason	SIP Status
No Bandwidth	480 Temporarily Unavailable
Gatekeeper Resource	404 Not Found
Unreachable Destination	404 Not Found
Destination Rejection	603 Decline
Invalid Revision	505 Version Not Supported
No Permission	401 Unauthorized
Unreachable Gatekeeper	503 Service Unavailable
Gateway Resource	480 Temporarily Unavailable

H.323 Disconnect Reason	SIP Status
Bad Format Request	400 Bad Request
Adaptive Busy	486 Busy Here
In Conference	486 Busy Here
Undefined Reason	500 Internal Server Error
Facility Call Deflection	486 Busy Here
Security Denied	401 Unauthorized
Called Party Not Registered	404 Not Found
Caller Not Registered	401 Unauthorized

IWF Termination During H.323 RAS

When a call that requires the IWF terminates from the H.323 side during RAS and generates an error, the Oracle Enterprise Session Border Controller uses the mapping scheme in the following table to determine the appropriate SIP status.

H.323 RAS Error	SIP Status
Called Party Not Registered	404 Not Found
Invalid Permission	401 Unauthorized
Request Denied	503 Service Unavailable
Undefined	500 Internal Server Error
Caller Not Registered	401 Unauthorized
Route Call To Gatekeeper	305 User Proxy
Invalid Endpoint ID	500 Internal Server Error
Resource Unavailable	503 Service Unavailable
Security Denial	401 Unauthorized
QoS Control Not Supported	501 Not Implemented
Incomplete Address	484 Address Incomplete
Route Call to SCN	302 Moved Temporarily
Aliases Inconsistent	485 Ambiguous
Not Currently Registered	401 Unauthorized

IWF RAS Registration Failure Code Mapping

For calls that require interworking between H.323 and SIP, the Oracle Enterprise Session Border Controller supports IWF response code mapping. This feature enables the Oracle Enterprise Session Border Controller to support configurable SIP response codes for IWF calls that fail during RAS, when the Oracle Enterprise Session Border Controller has been unable to register with a gatekeeper; this allows a wider range of more accurate response codes to be communicated.

When this feature is not enabled, the Oracle Enterprise Session Border Controller generates a 404 Not Found when a SIP-to-H.323 call fails as a result of the stack's failure to register with a gatekeeper.

When the condition noted above takes place, the response code can be any of the ones listed in this table. The code values listed in the table are used to specify the code to which you want to map.

Code	Description
403	Forbidden
406	Not Acceptable
408	Request Timeout
410	Gone
420	Bad Extension
480	Temporarily Unavailable
486	Busy Here
487	Request Terminated
500	Server Internal Error
503	Service Unavailable
504	Server Time-out
600	Busy Everywhere
603	Decline

To enable IWF RAS registration failure code mapping:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (iwfRegFailCode=X), and then press Enter. X is the SIP response code that you want to use; the table above lists the supported response codes that are supported.

```
ACMEPACKET(h323)# options +iwfRegFailCode=503
```

   If you type options iwfRegFailCode=X, you will overwrite any previously configured options. In order to append the option to the options list, you must prepend the new option with a plus sign as shown in the previous example.

IWF Termination from SIP

When a call that requires the IWF terminates from the SIP side, the Oracle Enterprise Session Border Controller uses the mapping scheme in the following table to determine the appropriate H.323 Release Complete Reason code.

SIP Status	H.323 Release Complete Reason
300 Multiple Choices	Undefined Reason
401 Unauthorized	Security Denied
402 Payment Required	Undefined Reason
403 Forbidden	No Permission

SIP Status	H.323 Release Complete Reason
404 Not Found	Unreachable Destination
405 Method Not Allowed	Undefined Reason
406 Not Acceptable	Undefined Reason
407 Proxy Authentication Required	Security Denied
408 Request Timeout	Adaptive Busy
409 Conflict	Undefined Reason
410 Gone	Unreachable Destination
411 Length Required	Undefined Reason
414 Request-URI Too Large	Bad Format Address
415 Unsupported Media Type	Undefined Reason
420 Bad Extension	Bad Format Address
480 Temporarily Unavailable	Adaptive Busy
481 Call/Transaction Does Not Exist	Undefined Reason
482 Loop Detected	Undefined Reason
483 Too Many Hops	Undefined Reason
484 Address Incomplete	Bad Format Address
485 Ambiguous	Undefined Reason
486 Busy Here	In Conference
487 Request Terminated	Undefined Reason
488 Not Acceptable Here	Undefined Reason
500 Internal Server Error	Undefined Reason
501 Not Implemented	Undefined Reason
502 Bad Gateway	Gateway Resource
503 Service Unavailable	Gateway Resource
504 Gateway Timeout	Adaptive Busy
505 Version Not Supported	Invalid Revision
600 Busy Everywhere	Adaptive Busy
603 Decline	Destination Rejection
604 Does Not Exist Anywhere	Unreachable Destination
606 Not Acceptable	Undefined Reason

Q.850 Cause to H.323 Release Complete Reason

When a call that requires the IWF terminates from the H.323 side and no H.323 Release Complete Reason is specified, the Oracle Enterprise Session Border Controller maps the Q.850 cause to an H.323 Release Complete Reason using the mapping scheme in the following table. This new H.323 status is then mapped to a SIP status as described in the IWF Termination from SIP table.

Q.850 Cause	H.323 Release Complete Reason
No Route To Destination	Unreachable Destination
Normal Call Clearing	Destination Rejection
User Busy	In Conference
Subscriber Absent	Called Party Not Registered
Invalid Number Format	Bad Format Address
Normal Unspecified	Undefined Reason
No Circuit/Channel Available	No Bandwidth
Network Out Of Order	Unreachable Gatekeeper
Temporary Failure	Adaptive Busy
Switching Equipment Congestion	Gateway Resource
Resource Unavailable	Gatekeeper Resource
Incompatible Destination	Invalid Revision
Interworking Unspecified	No Permission

Codec Mapping

The Oracle Enterprise Session Border Controller uses the following mapping scheme when converting media specifications between H.245 (used in H.323) and SDP (used in SIP).

Media coming into the Oracle Enterprise Session Border Controller one way exits the system in the corresponding way as specified in the following table. For example, media coming into the Oracle Enterprise Session Border Controller as H.245 type g711Ulaw64k exits the system as media type PCMU.

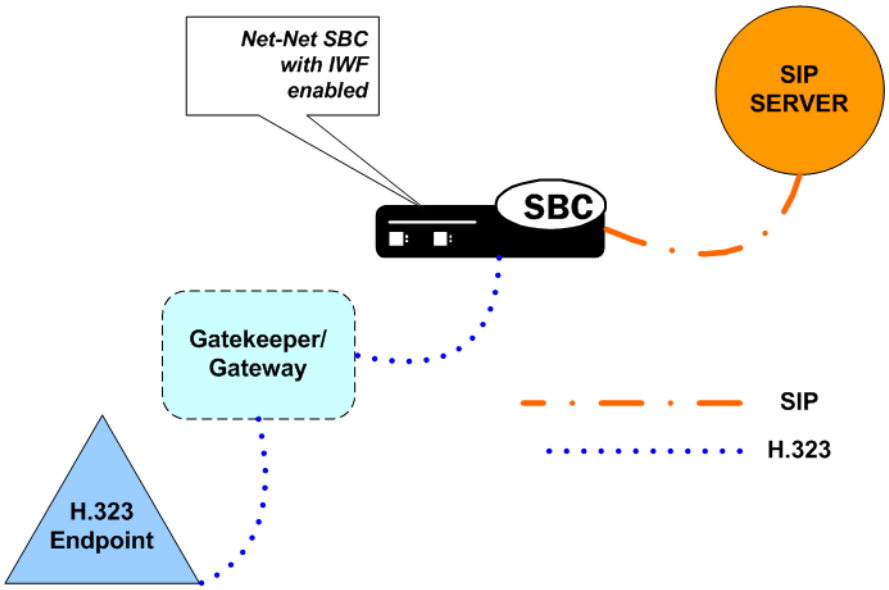
H.245 Type	SDP Media Type
g711Ulaw64k	PCMU
g711Ulaw56k	PCMU
g711Alaw64k	PCMA
g711Alaw56k	PCMA
g726	G726-32
g7231	G723
g722	G722
g728	G728
g729wAnnexB	G729
g729	G729 fmtp:18 annexb=no
h261VideoCapability	H261
h263VideoCapability	H263

IWF Service Enhancements

This section describes the Oracle Enterprise Session Border Controller features that are supported for when the Oracle Enterprise Session Border Controller performs interworking between SIP and H.323. Enabling these enhancements only requires that you set up a fully functional SIP configuration, a fully functional H.323 configuration, and that you enable IWF on your Oracle Enterprise Session Border Controller . You do not have to set any special configuration because these enhancements happen automatically.

SIP Redirect—H.323 LRQ Management

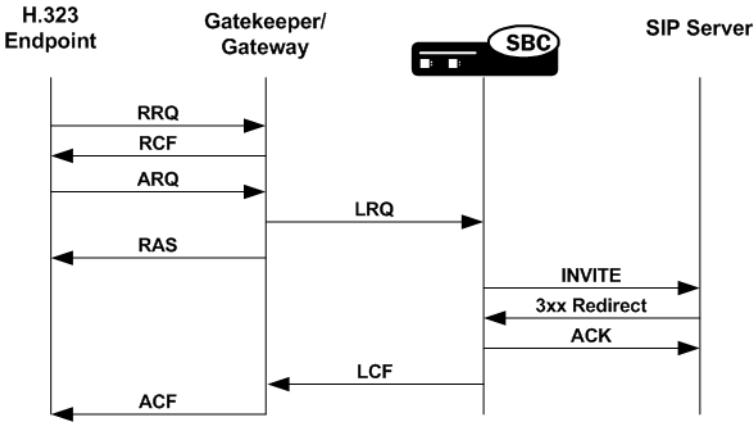
When it needs to interact with a SIP Redirect server, the Oracle Enterprise Session Border Controller can interpret the SIP messages and manage them on the H.323 side of the session. For IWF sessions, the Oracle Enterprise Session Border Controller handles SIP Redirect and H.323 LRQ messages.



Redirect—LRQ Management Sample 1

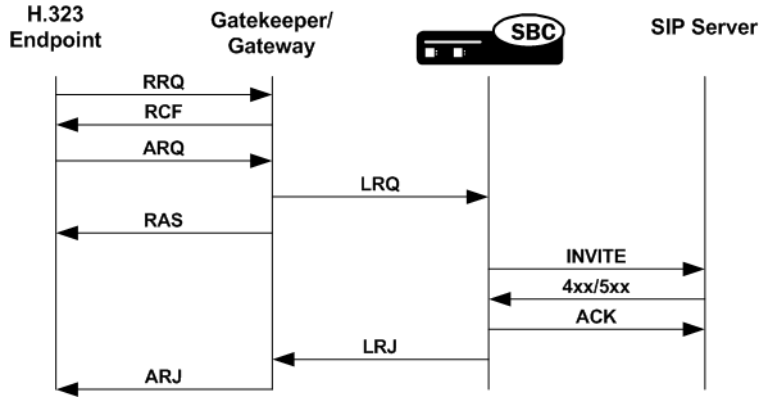
This section presents three possible scenarios for SIP Redirect-H.323 LRQ management.

The following diagram shows an established session that uses SIP Redirect—H.323 LRQ management. Here, the Oracle Enterprise Session Border Controller sends an INVITE to a SIP Redirect Server that responds with a 3xx Redirection message. The Oracle Enterprise Session Border Controller then sends the gatekeeper/gateway an LCF message that causes an ACF message to be sent to the H.323 endpoint.



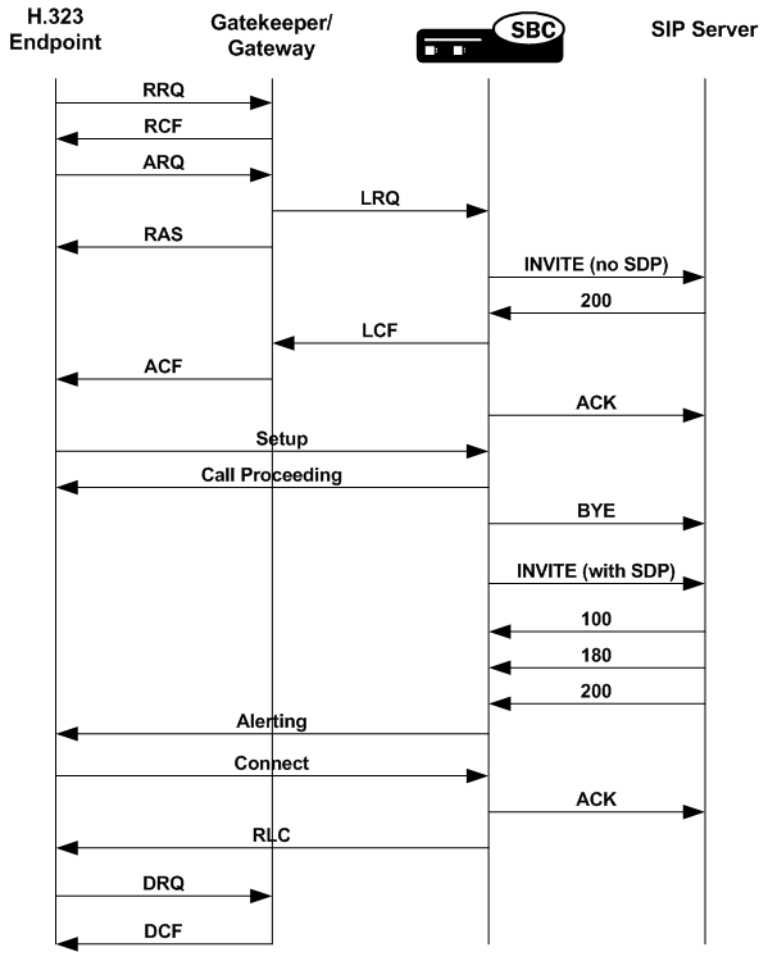
Redirect—LRQ Management Sample 2

The following diagram shows how the Oracle Enterprise Session Border Controller handles the exchange when the SIP Redirect server declares either that there is an error or that there is no such user. These SIP messages come from either the 4xx Request Failure or 5xx Server Failure series. In the example below, the SIP Redirect server returns a 401 Unauthorized message, which the Oracle Enterprise Session Border Controller interworks and communicates to the H.323 gatekeeper as an LRJ. Then the H.323 gatekeeper/gateway issues an ARJ to the H.323 endpoint.



Redirect—LRQ Management Sample 3

In this call flow, the SIP server issues a 2xx Successful message that is not supposed to be sent (because a 3xx, 4xx, or 5xx message should be sent in response to the Oracle Enterprise Session Border Controller ’s INVITE). The Oracle Enterprise Session Border Controller sends a BYE message to the SIP Redirect Server, but it tries to initiate the session again, this time successfully. The final sample call flow shown rarely occurs.



SIP INFO and DTMF UII Management

The Oracle Enterprise Session Border Controller supports DTMF for that require the IWF, enabling features such as keypress, alphanumeric, and hookflash. Because tones are not transmitted as audio, they must pass as out-of-band signaling information, meaning that the Oracle Enterprise Session Border Controller needs to convert an H.245 UII (User Input Indication) into SIP.

Depending on the capability of the H.323 endpoint, the Oracle Enterprise Session Border Controller sends either an alphanumeric or DTMF signal in the H.245 UII. The Oracle Enterprise Session Border Controller sends nothing if the endpoint does not support an alphanumeric or DTMF signal. The SIP INFO message will have a content type of application/dtmf-relay, and the message body will be in the form Signal=*\r\nDuration=250\r\n. If the duration is absent in the SIP INFO or the UII received on the H.323 side is alphanumeric, the Oracle Enterprise Session Border Controller uses the a 250 millisecond default value.

Mid-Session Media Change

Mid-session media change happens during a call that requires the IWF when the type of media being sent while a session is in progress changes. For example, a fax transmission might require mid-session media change; besides fax, other applications of this feature are possible. To support the transmission of a T.38 fax sent over an IWF session, some media channels must be opened and others closed. In addition, the Oracle Enterprise Session Border Controller can accommodate a request for media change from, for example, audio to an image type for T.38 fax.

Because the media requirements are driven by endpoints and Gateways, you do not have to configure the Oracle Enterprise Session Border Controller s mid-session media change support.

Enhanced Support for FAX Calls

The Oracle Enterprise Session Border Controller now supports T.38 fax calls in networks containing elements that do not comply with the ITU-T H.323 Annex D recommendation for how to replace an existing audio stream with a T.38 fax stream. This support applies to signaling that requires interworking between SIP and H.323.

In the standard call model following the ITU-T recommendation, the endpoint detecting the fax tone sends an H.245 RequestMode message to its peer with a T.38 data mode. The receiving endpoint returns a RequestMode Ack by way of acknowledgement, triggering the sending endpoint to close its audio channel and open a T.38 fax channel. The receiving endpoint closes and opens the same channels on its end. T.38 fax streams flow upon the acknowledgement of all relevant channels.

However, certain endpoints close their logical channel before sending the H.245 RequestMode message for T.38, leaving the Oracle Enterprise Session Border Controller with its audio channel still open and without having attempted to open a T.38 fax channel. To overcome this issue, the Oracle Enterprise Session Border Controller now checks whether or not audio channels have been closed whenever it receives an H.245 RequestMode message for T. 38. If it finds a closed audio channel, the Oracle Enterprise Session Border Controller checks for the presence of a matching outgoing audio channel. A match causes the Oracle Enterprise Session Border Controller to close the audio channel and continue with the procedure for converting to T.38 fax.

Removing the T.38 Codec from an H.245 TCS

For SIP-H.323 IWF sessions, H.323 automatically inserts the T.38 FAX codec in the H.246 TCS message. You can stop this insertion using the remove-t38 parameter in the H.323 global configuration.

Early Media

For call that require the IWF, the Oracle Enterprise Session Border Controller supports a cut-through for early media for calls that originate in SIP or H.323.

For a session originating in SIP, the provisional message will contain the SDP information if a Fast Start OLC was received in the Call Proceeding, Alerting, or Progress messages. The same SDP will be sent in the SIP 200 OK.

For a session that starts in H.323, the Oracle Enterprise Session Border Controller translates the SDP it receives in SIP messages (either a 180 or a 183) into the appropriate H.323 Fast Start elements: Alerting or Progress. If the Alerting or Progress messages contain Fast Start elements, the Progress Indicator Q.931 information element (IE) will also be included in the message with Progress Descriptor 8, indicating that in-band information or an appropriate pattern is now available. This causes the call party to enable end-to-end early media in the reverse direction in accordance with H.323 v4.

In addition, the Oracle Enterprise Session Border Controller allows early media to flow in the forward direction for a call that requires the IWF starting in H.323 that is being translated to SIP. This happens after the Oracle Enterprise Session Border Controller has received provisional response with SDP and has sent Alerting or Progress message with Fast Start to the calling party. Similarly, early media in the forward direction is enabled for a call that requires the IWF starting in SIP and being translated to H.323. This happens after the Oracle Enterprise Session Border Controller received Alerting or Progress messages with Fast Start and maps the Alerting or Progress to SIP 180 or 183 provisional response with the SDP answer.

Display Name Mapping

The Oracle Enterprise Session Border Controller displays the full name and number of the calling party (for features such as Caller ID) when it handles calls that require the IWF. The Oracle Enterprise Session Border Controller takes the display name in the From field of the SIP INVITE and maps it to the display IE so that it can show the full name of the calling party.

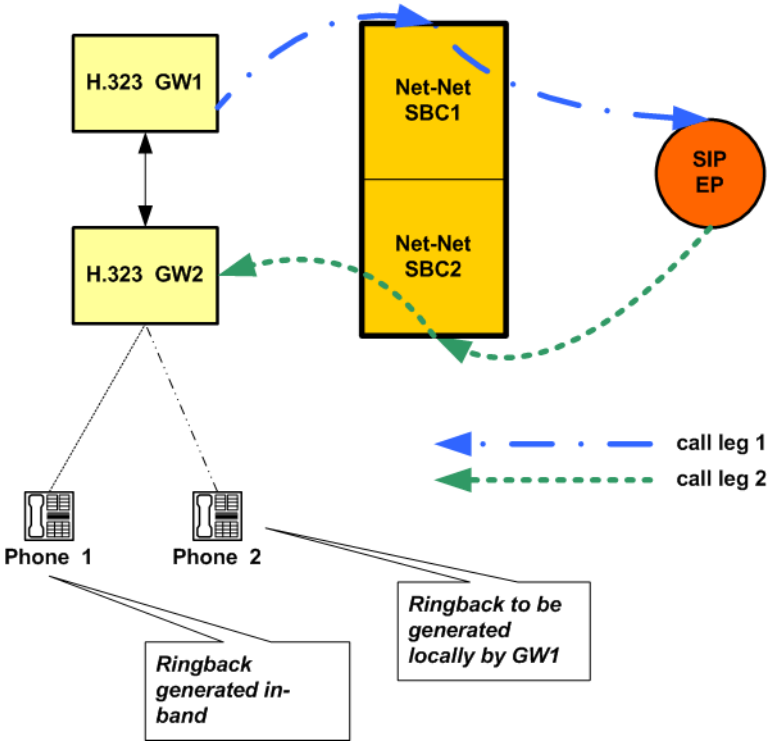
IWF Ringback Support

When interworking SIP and H.323 to a gateway, PSTN gateway, or other endpoint, the Oracle Enterprise Session Border Controller uses the mappings shown in the table below. The absence or presence of SDP in the SIP provisional message determines whether the tones are generated in-band or locally.

For each of the mappings listed in the following table, this section provides a sample call flow.

SIP Message	H.323 Message
No Message	CallProceeding
No Message	Progress without PI
183 with SDP	Progress with PI
180 w/o SDP	Alert without PI
180 with SDP	Alert with PI

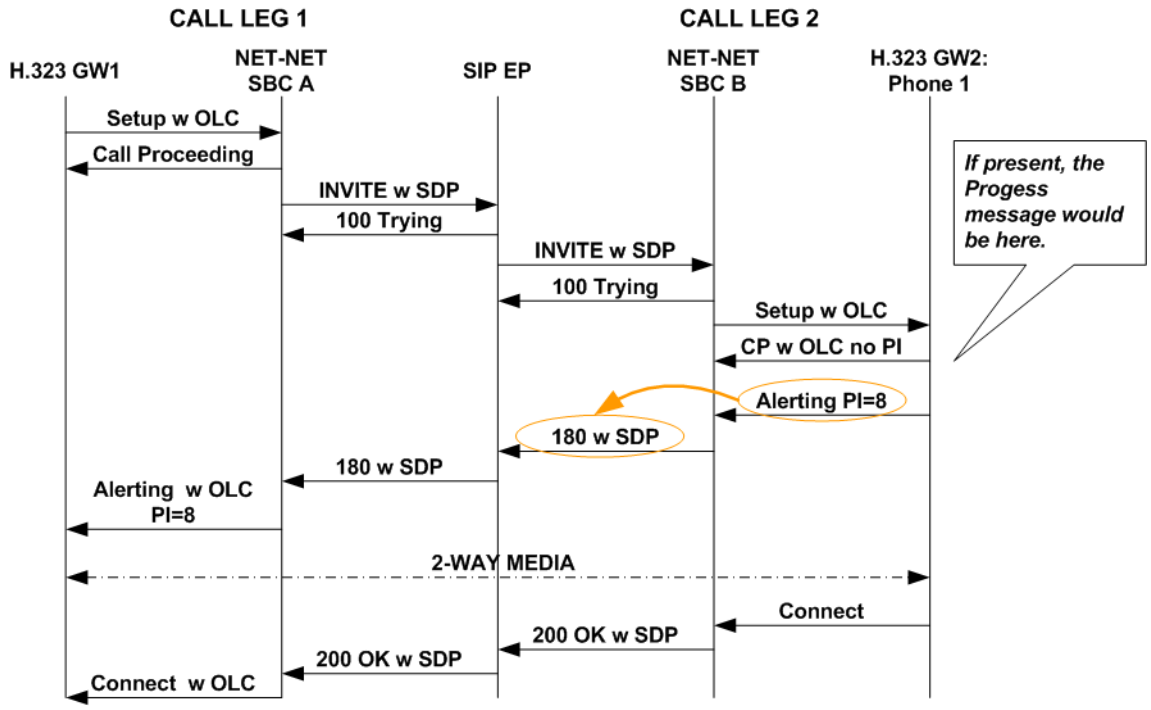
In the following diagram, a call that requires the IWF passes through the Oracle Enterprise Session Border Controller twice, creating two call legs. The call originates from H.323 GW1 and terminates in Phone 1 or Phone 2.



Sample 1 In-band Ringback without Progress Message

This sample flow shows how the Oracle Enterprise Session Border Controller handles a call that requires the IWF where there is no progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

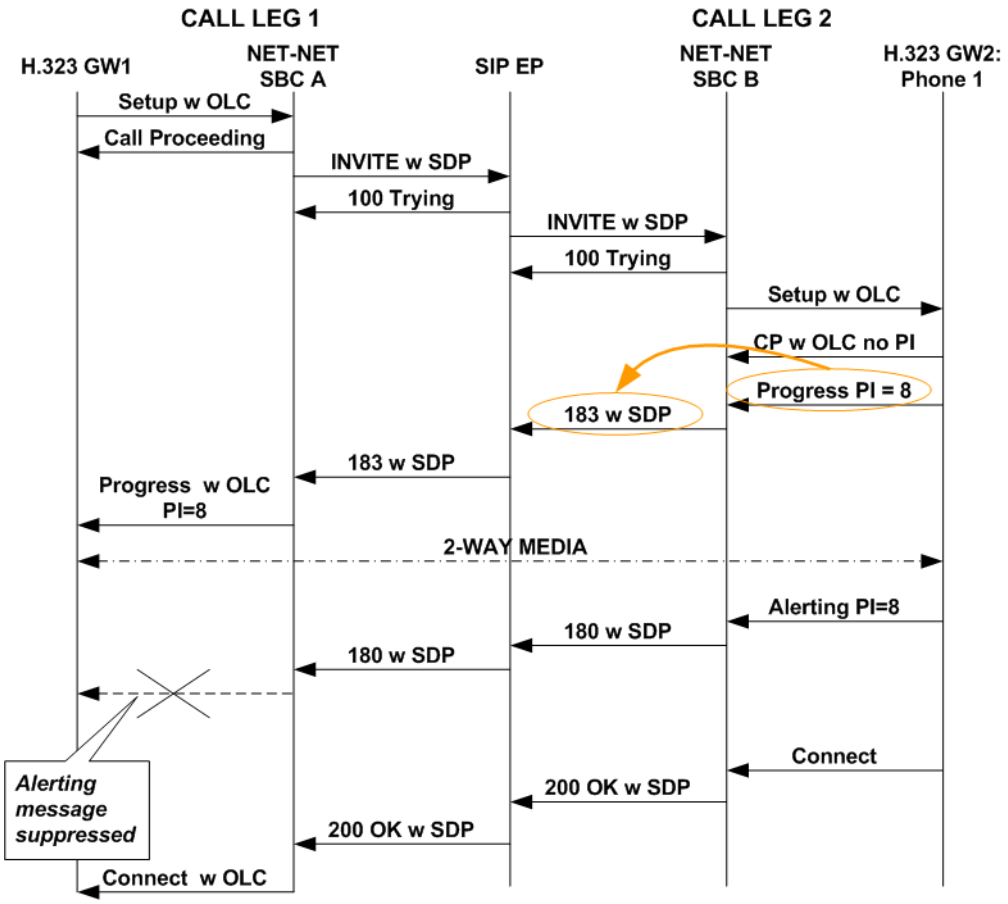
In this diagram, you can see that the Oracle Enterprise Session Border Controller maps the progress indicator included in the Alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. When the Progress message appears, it contains the progress indicator rather than the Alerting message containing it.



Sample 2 In-band Ringback with Progress Message

This sample flow shows how the Oracle Enterprise Session Border Controller handles a call that requires the IWF where there is a progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

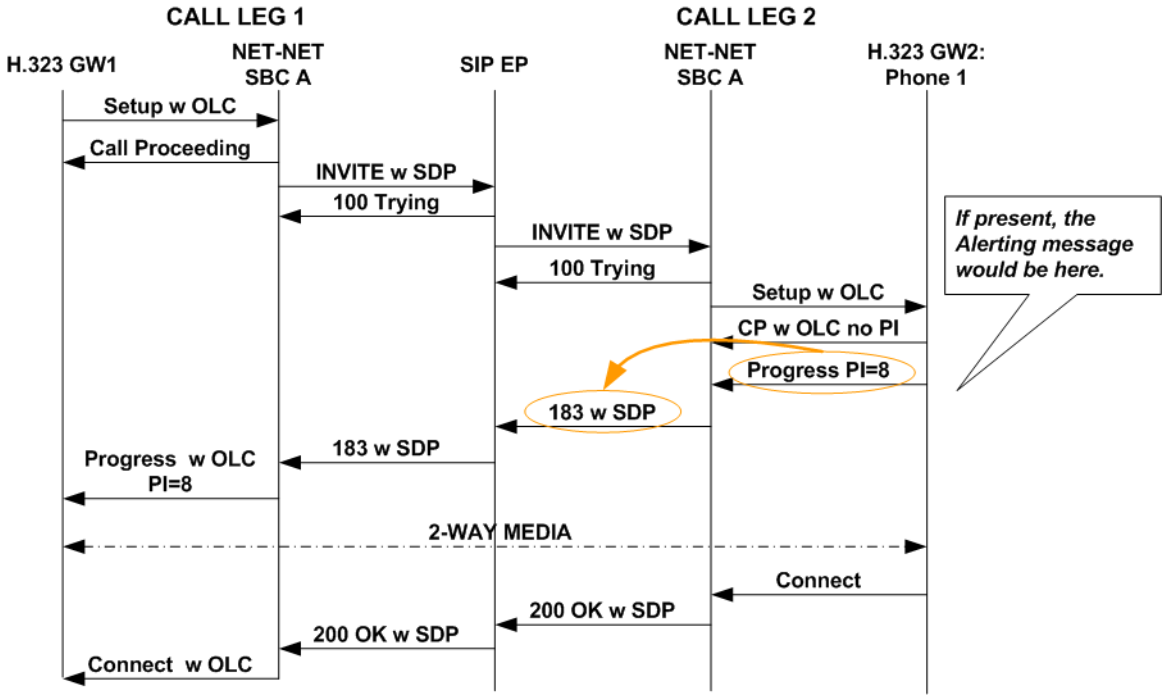
For this call flow, you can see again that the Oracle Enterprise Session Border Controller maps the progress indicator included in the alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. Note that now the Progress message contains the progress indicator.



Sample 3 In-band Ringback without Alerting Message

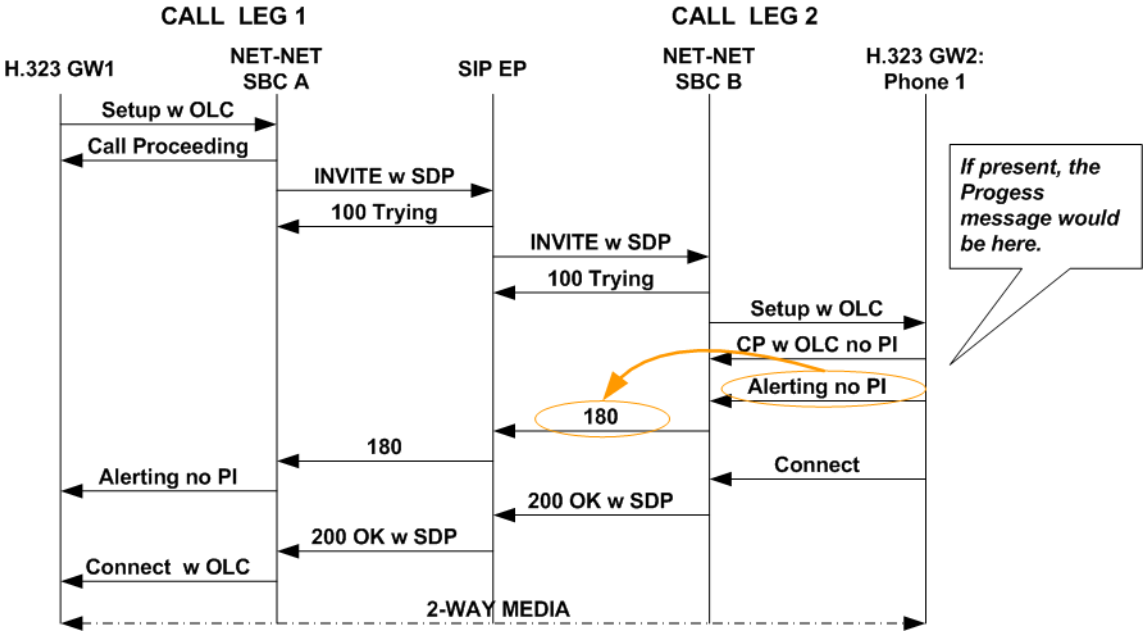
This sample flow shows how the Oracle Enterprise Session Border Controller handles a call that requires the IWF where there is no progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

In this diagram, you can see that the Oracle Enterprise Session Border Controller maps the progress indicator included in the Progress message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. When the Alerting message appears, it contains the progress indicator rather than the Progress message containing it.



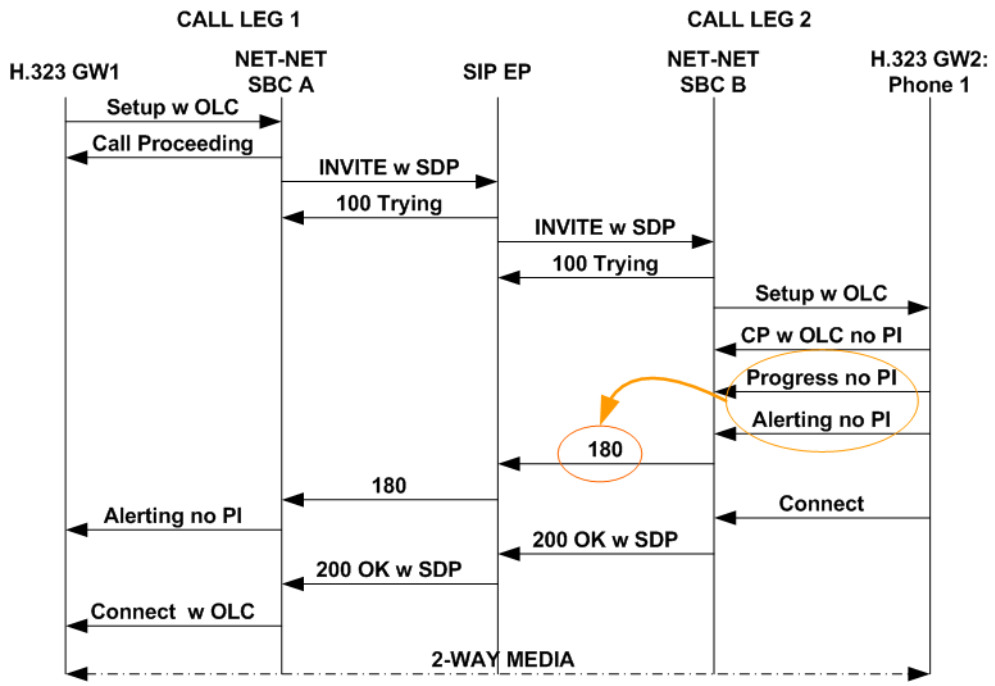
Sample 4 Out-of-band Ringback without Progress Message

When there is no progress indicator included in the Alerting message, then there is out-of-band ringback. The system maps the Alerting message to a SIP 180, but it it does not include SDP in the SIP 180. This call flow shows that there is no Progress message and that media cannot be set up until after H.323 Connect and SIP messages are sent.



Sample Flow 5 Out-of-band Ringback with Progress Message

When there is no progress indicator included in either the Alerting or Progress messages, then there is out-of-band ringback. The system maps the Alerting message to a SIP 180, but it does not include SDP in the SIP 180. This call flow shows includes the Progress message; still, media cannot be set up until after H.323 Connect and SIP messages are sent.

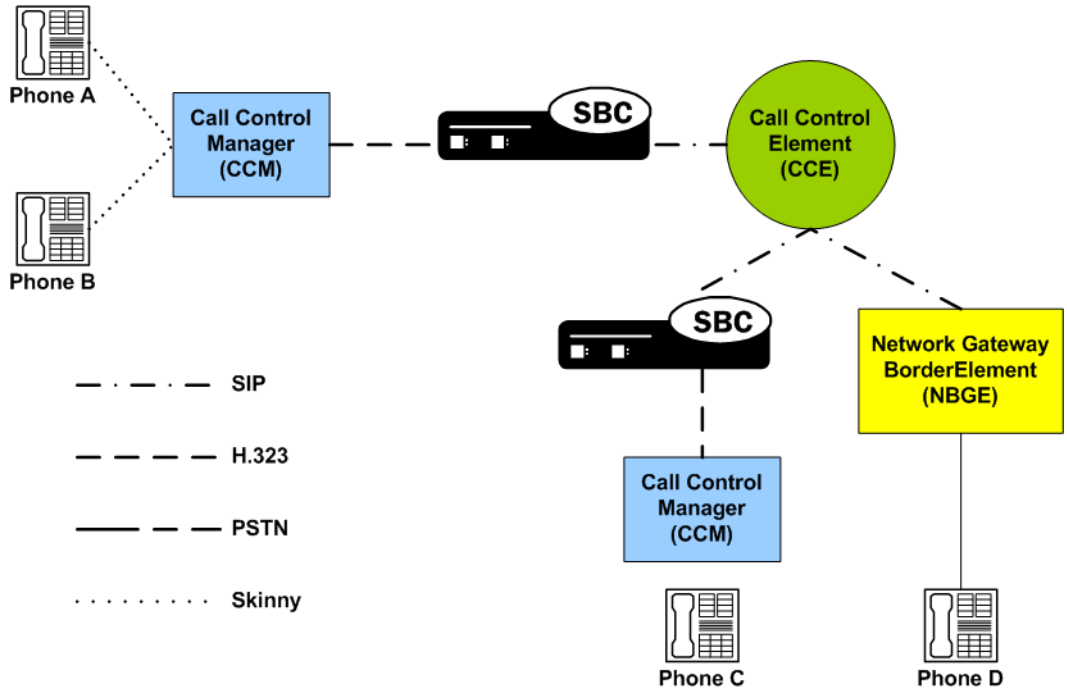


H.323 Endpoint-Originated Call Hold and Transfer

When calls that require the IWF originating in H.323, the Oracle Enterprise Session Border Controller supports call hold, transfer, and conference for the H.323 call leg. The call hold and transfer feature uses signaling procedures based on the ITU-T recommendations/H.323 specification for third party initiated pause and rerouting.

You do not have to configure the Oracle Enterprise Session Border Controller ’s call hold and transfer feature.

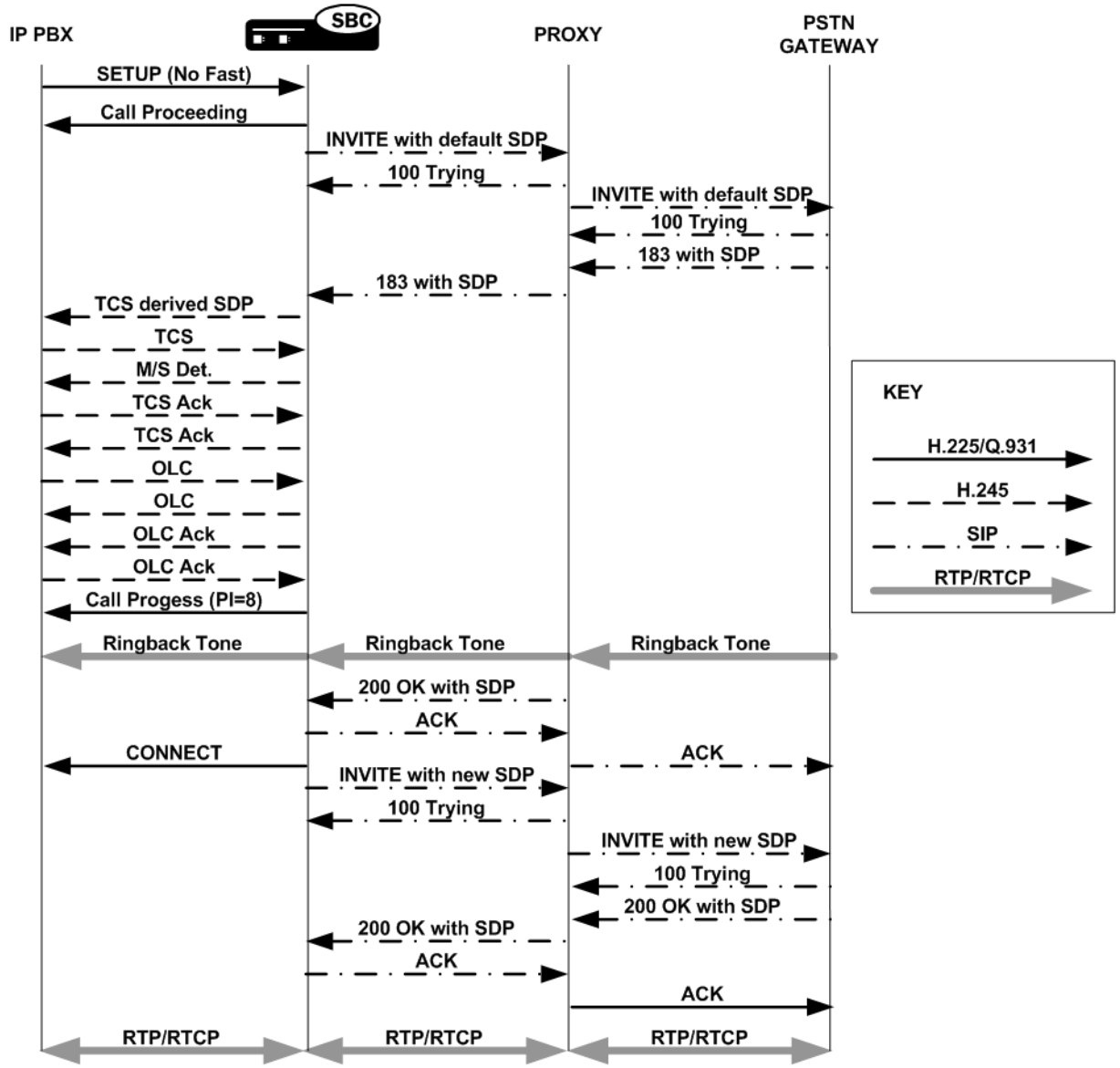
The following diagram shows how the Oracle Enterprise Session Border Controller provides call hold and transfer support for IWF sessions that originate in H.323. As you review this section’s call flow diagrams, you might want to refer back to the following logical diagram directly below to review the network elements involved, and what protocols they use.



Basic Call

In the following sample basic call, IP PBX A sends an H.323 Slow Starts message ultimately destined for the PSTN through the Oracle Enterprise Session Border Controller . The Oracle Enterprise Session Border Controller performs translation to SIP and inserts default information about media. Once the PSTN gateway responds with a 183 containing SDP, the Oracle Enterprise Session Border Controller sends that information to IP PBX A. Then the Oracle Enterprise Session Border Controller and the IP PBX exchange TCS- and OLC-related messages, and they negotiate master-slave determination. The Oracle Enterprise Session Border Controller also sends IP PBX A a Call Progress message with a progress indicator of 8.

After the ringback tone, the proxy sends a 200 OK message with SDP to the system. The Oracle Enterprise Session Border Controller sends a Connect message to the IP PBX A, and then it sends another SIP INVITE to the proxy that contains amended SDP (if that information about media is different from the default). After 200 OK and ACK messages are exchanged, media (RTP/RTCP) flow takes place.

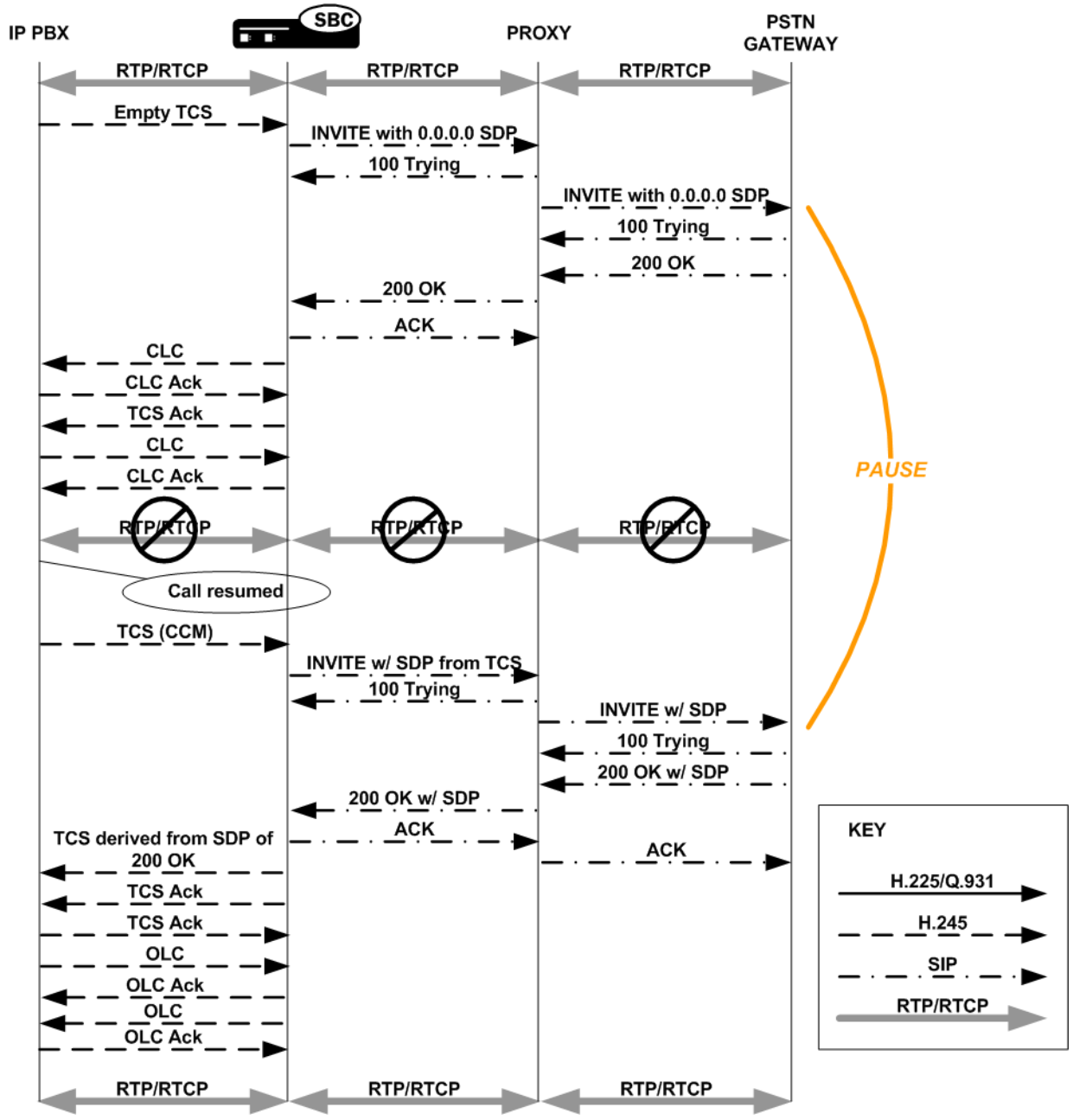


Hold

This sample call flow assumes that the IWF call is established and that the RTP/RTCP flow is already in progress. The hold button is pushed, and IP PBX A sends an empty TCS to the Oracle Enterprise Session Border Controller . The Oracle Enterprise Session Border Controller puts the called party on hold by sending an INVITE message with 0.0.0.0 SDP to the SIP side of the call. Using 0.0.0.0 as the media address effectively stops the media flow. This

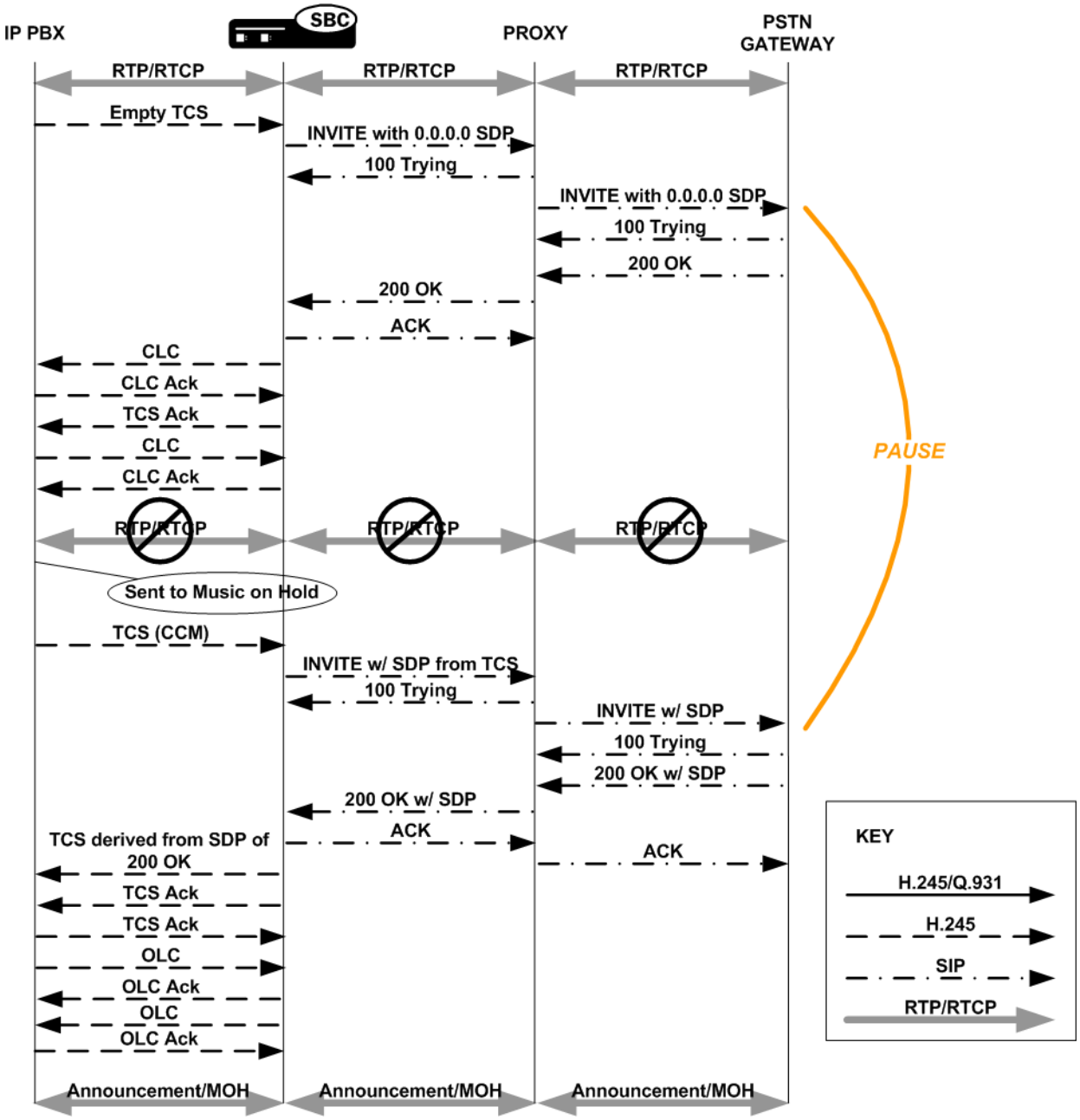
INVITE is acknowledged, and the Oracle Enterprise Session Border Controller closes the channels on the H.323 side, halting the RTP/RTCP flow.

When the caller on the H.323 side takes the call off hold, it resumes with a TCS that the Oracle Enterprise Session Border Controller receives and then translates on the SIP side as an INVITE with SDP. After that INVITE is acknowledged and received, the Oracle Enterprise Session Border Controller opens logical channels on the H.323 side and RTP/RTCP flows resume.



Music On Hold

This scenario is similar to the hold feature enabled for calls that require the IWF, except that after the RTP/RTCP flow between the H.323 and SIP sides stops, the call is sent to music on hold. Before the announcement or music plays, the Oracle Enterprise Session Border Controller sets up the necessary support for media to be exchanged.



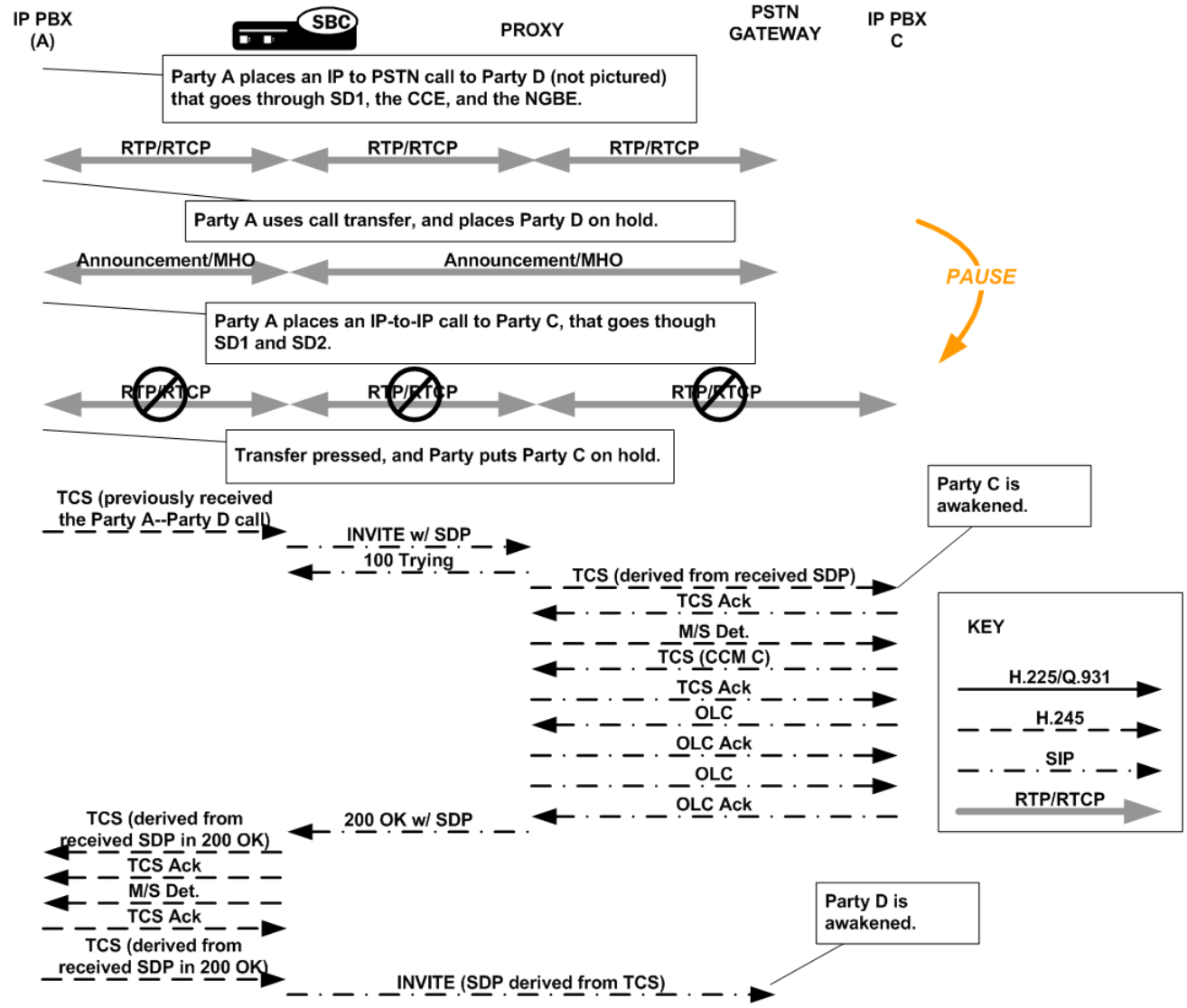
Transfer

The call flow described in this section recalls the diagram at the top of the H.323 Endpoint-Originated Call Hold and Transfer section, where endpoints A, B, and C are H.323 devices and endpoint D is a SIP device. When you follow the signaling and media flows, note that there are two Oracle Enterprise Session Border Controller s in the call transfer and two sets of SIP/H.323 translations that take place. The first Oracle Enterprise Session Border Controller translates H.323 to SIP, and the second performs the same operations with the protocols reversed.

In the scenario pictured, Party A is on a call with Party D, but wants to transfer Party C to Party D. Party A places Party D on hold, and then makes the call to Party C. Party A then puts Party C on hold, pressing the transfer button. You can see that Oracle Enterprise Session Border Controller 1 receives a TCS from the IP PBX, which is then translated to SIP. Oracle Enterprise Session Border Controller 2 receives it, performs the required protocol translations, and then opens a session with Party C via another IP PBX. Once this session is up and Party D is awakened, channels are established for media exchange.

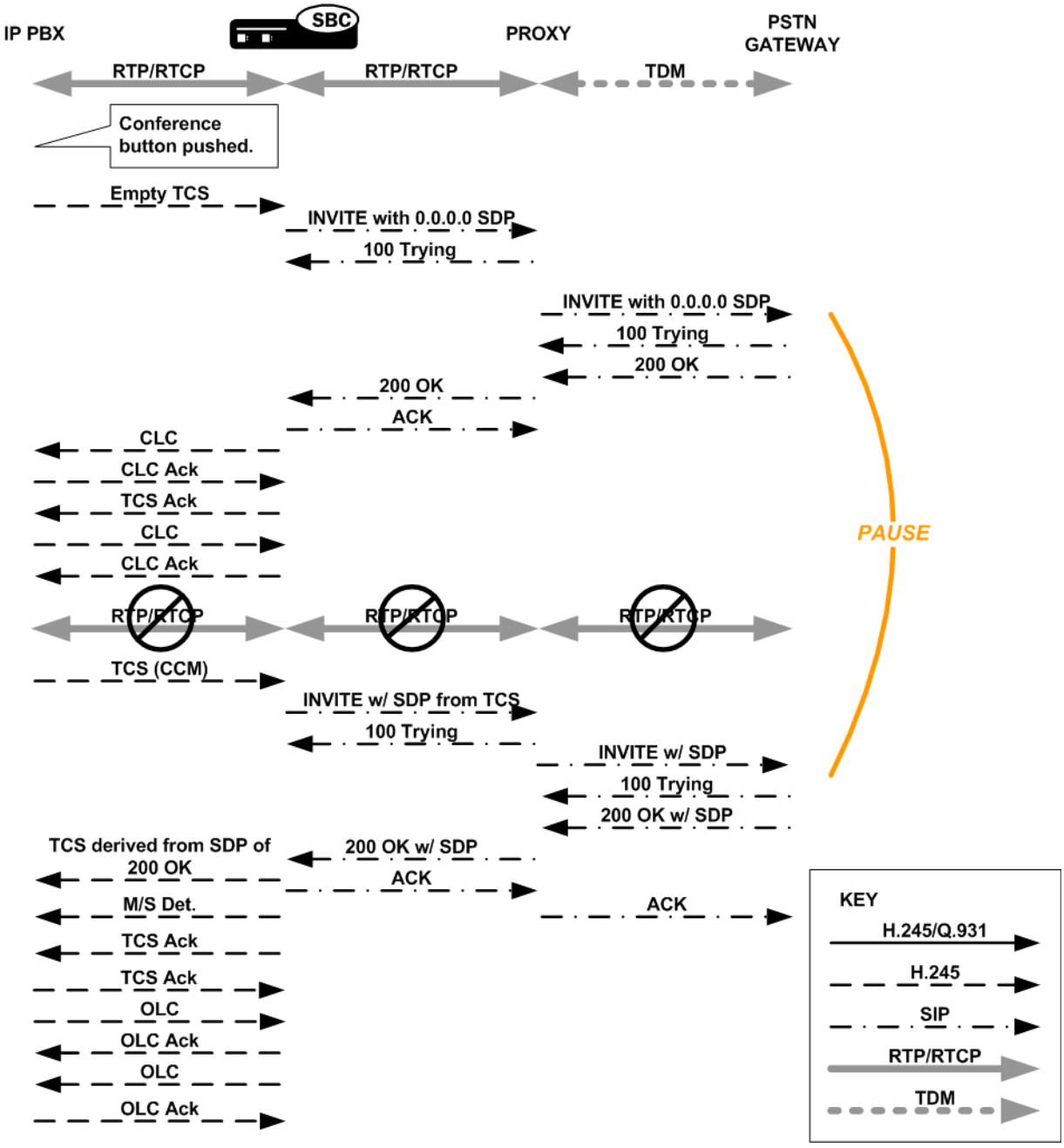
In order to redirect the media so that it flows between Party C and Party D, the Oracle Enterprise Session Border Controller 1 and IP PBX C exchange OLC and OLC Ack messages that contain address information for Party C and for Party D. Address information for both parties is contained in the OLC Ack messages that the Oracle Enterprise Session Border Controller exchanges with the IP PBX. IP PBX A does not move forward with the call until it has the necessary address information.

Even though Party A's participation in the call stops early in this scenario, the IP PBX with which it is associated keeps the signaling sessions with the Oracle Enterprise Session Border Controller alive to manage the transfer.



Conference

To conference a call that requires the IWF that starts in H.323, the Oracle Enterprise Session Border Controller uses a scenario much like the one used for holding a call that requires the IWF. Here again, the INVITE with 0.0.0.0 as the media address and the closing of logical channels stops the flow of RTP/RTCP. After signaling and SDP/media information are re-established, RTP/RTCP for the conference flows.



IWF Call Forwarding

This section describes the Oracle Enterprise Session Border Controller 's IWF Call Forwarding feature, which is supported for calls initiated in SIP that require interworking to H.323.

Prior to the implementation of this feature, the Oracle Enterprise Session Border Controller did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

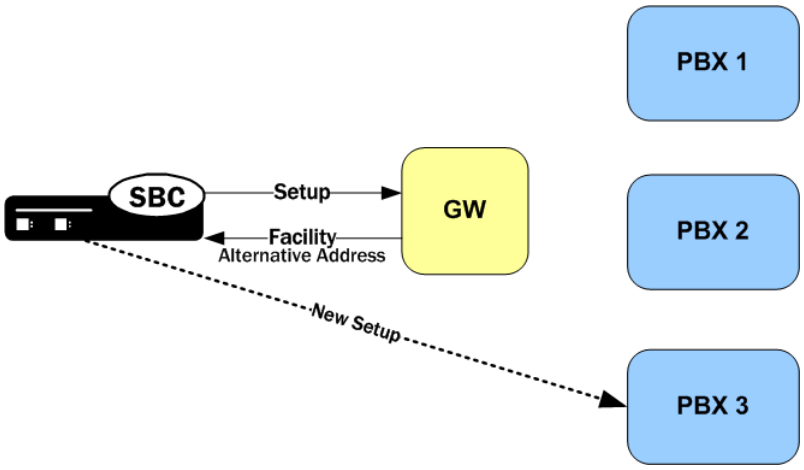
- Fail to release the initial call and initiate the forwarded call
- Drop the entire call when the remote endpoint for the call tore down the session

New Behavior

In the diagram below, you can see that the Oracle Enterprise Session Border Controller sends the initial Setup message to the gateway, and the gateway returns the Facility message with an alternate address for forwarding. Rather

than engaging in its former behavior, the Oracle Enterprise Session Border Controller now releases the call with the gateway and sends a new Setup to the alternate address from the Facility message.

This new Setup up has no effect on the first call leg, which remains connected.



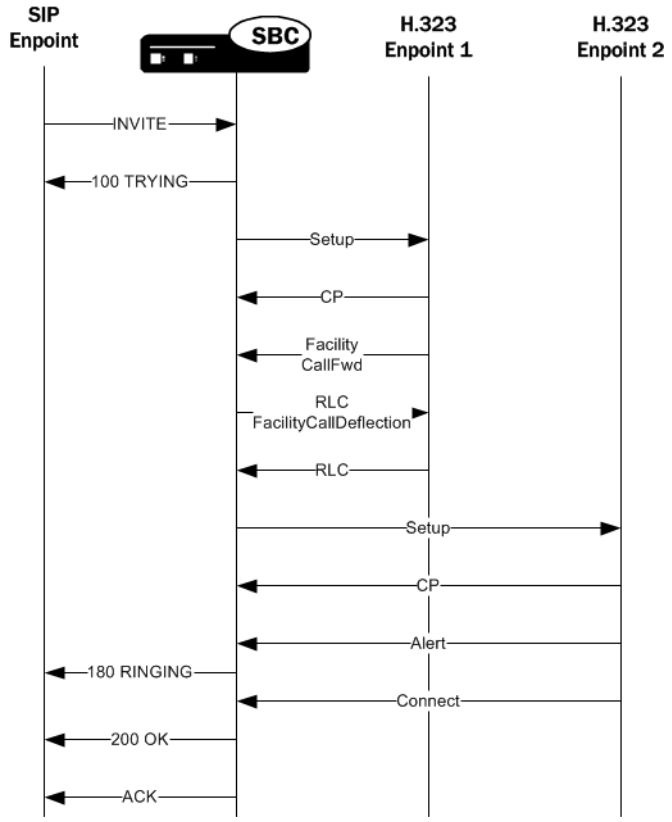
When it receives a Facility message with the reason CallForwarded, the Oracle Enterprise Session Border Controller looks for an alternate transport address in the Facility's alternativeAddress or alternativeAliasAddress element. The Oracle Enterprise Session Border Controller releases the egress call with the reason facilityCallDeflection. Then it takes one of two courses of action:

- If it does not find an alternative address, the Oracle Enterprise Session Border Controller releases the ingress call (with 486 BUSY HERE for a call being interworked from SIP to H.323).

If it finds an alternative address and the egress call has not been alerted or answered, the Oracle Enterprise Session Border Controller at this point tries to initiate a new egress call. The Oracle Enterprise Session Border Controller uses the alternative alias address to populate the calledPartyNumber information element (IE) and the destination address of the new Setup.

H.323 Sample Call Flow

The following diagram shows how the H.323 Call Forwarding feature works in a purely H.323 environment.



Media Release for H.323 SS-FS Calls for IWF

When the Oracle Enterprise Session Border Controller routes a slow-start to fast-start call, it is possible for the same fast-start call to be routed back through the Oracle Enterprise Session Border Controller making for a hairpin flow. If it does becomes a hairpin flow, then the Oracle Enterprise Session Border Controller routes it to its destination as a fast-start to fast-start call. This can result in one-way media if:

- The destination of the hairpin call is in the same realm as the originating slow-start to fast-start call
- The realm reference in the first bullet item is configured to disable in-realm media management
- The called endpoint accepts the proposed fast-start logical channels

The enhancements to the Oracle Enterprise Session Border Controller ’s behavior described in this section show how the Oracle Enterprise Session Border Controller follows additional procedures when setting up a hairpin flow to avoid one-way media when media release occurs.

H.323

For H.323 calls, the Oracle Enterprise Session Border Controller establishes media using the H.245 procedures described in the H.245 ITU-T recommendation: control protocol for multimedia communication. It also uses the Fast Connect procedure defined in the H.323 ITU-T recommendation: packet-based multimedia communication systems.

The latter ITU-T recommendation allows a calling endpoint to send a Setup message that contains a fastStart element, a sequence of OLC structures that describe the calling endpoint’s proposed forward/reverse logical channels. If the called endpoint accepts this proposal, then logical channels are established.

When the Oracle Enterprise Session Border Controller translates a call originating in slow-start to fast-start, it uses a Fast Connect procedure in the outgoing leg by sending an outgoing Setup that includes a fastStart element with one or more OLC structures. But when the Oracle Enterprise Session Border Controller constructs this message, it is unaware of whether the call will become hairpinned or if media release will occur. Because it does not yet have this information, the Oracle Enterprise Session Border Controller sets the Network Address and the TSAP identifier in the OLC structures to the ingress IP address and port of a corresponding media flow allocated for media traveling between the calling and called endpoints. So if the called endpoint accepts the fastStart the Oracle Enterprise Session

Border Controller proposes, the called endpoint would send its media to the Oracle Enterprise Session Border Controller . After acceptance, the system starts H.245 procedures on the slow-start side of the call to set up logical channels on that side. Then the Oracle Enterprise Session Border Controller updates the IP address and port of the media flows using OLC and OLCAck messages received from the calling endpoint.

This procedure works well for endpoints that are not in the same realm, or that are in the same realm for which media management is disabled, because each endpoint must send its media through the Oracle Enterprise Session Border Controller . When the endpoints are in the same realm and when media management is enabled, however, the Oracle Enterprise Session Border Controller must perform additional steps for media release in slow-start to fast-start calls.

To support media release in slow-start to fast-start calls, the Oracle Enterprise Session Border Controller performs a hold-and-resume procedure on the fast-start side. After it establishes channels on the slow-start side and if it detects media release being enabled, the Oracle Enterprise Session Border Controller sends and empty TCS to the fast-start side to put that side on hold. Then the called endpoint closes all the logical channels it previously opened in the Fast Connect procedure and stops transmitting to them. And the Oracle Enterprise Session Border Controller also closes it logical channels. Once the channels are closed, the Oracle Enterprise Session Border Controller resumes the call by sending a new, restricted TCS to the fast-start side. The restricted TCS only contains the receive and transmit capabilities of the codecs types that the called endpoint accepted in the Fast Connect procedure, and it forces the called endpoint to re-open logical channels of the same codec types accepted in the Fast Connect procedure. Once it receives and OLC from the called endpoint, the Oracle Enterprise Session Border Controller sends on OLCAck with the Network Address and TSAP identifier for the logical channel from the calling endpoint. Then the Oracle Enterprise Session Border Controller re-opens logical channels (of the same codec types that it open in the Fast Connect procedure). If the called endpoint has not changed its Network Address and TSAP identifier for its logical channels, media is re-established after the Oracle Enterprise Session Border Controller and the called endpoint exit the hold state. The last steps is for the Oracle Enterprise Session Border Controller to re-sends the full TCS message from the calling to the called endpoint to inform the called endpoint of the full capabilities of the calling endpoint.

Hold-and-Resume Procedure

The hold-and-resume procedure has three states:

- Media Hold—Starts when the Oracle Enterprise Session Border Controller sends the empty TCS to the called endpoint to put it on hold.

  When it detects media release, the Oracle Enterprise Session Border Controller puts the called endpoint on hold. It can only do so if it has exchanged the TCS/TCSAck messages and completed master-slave determination with the calling endpoint.

  When theOracle Enterprise Session Border Controller receives a TCSAck in response to the empty TCS that it sent to the called endpoint, it closes the logical channels it opened as part of the Fast Connect procedure; the called endpoint likewise closes its logical channels. The two then exchange CLC and CLCAck messages, which signals the start of the Media Resume state.
- Media Resume—Starts when the Oracle Enterprise Session Border Controller sends a restricted TCS to resume the call.

  The restricted TCS the Oracle Enterprise Session Border Controller sends contains only the receive/transmit capabilities of the codec types previously accepted by the called endpoint in the Fast Connect procedure. This forces the called endpoint to re-open logical channels of the same codec type that were previously accepted in the Fast Connect procedure.

  After sending this TCS, the Oracle Enterprise Session Border Controller is ready (as specified in the ITU-T recommendations) to take part on the master-slave determination (MSD) process. However, the called party and not the Oracle Enterprise Session Border Controller initiates the MSD if it is required. The MSD is completed if necessary. Alternately, the called endpoint can start to re-open its logical channels. When it receives the first OLC from the called endpoint, the Oracle Enterprise Session Border Controller also starts to re-open its logical channels.
- Media Complete—Starts when all the logical channels that the Oracle Enterprise Session Border Controller re-opens are acknowledged by the called endpoint.

When it enters the Media Complete state, the Oracle Enterprise Session Border Controller updates the called endpoint with the full capabilities of the calling endpoint by sending the full TCS.

Additional IWF Steps

For calls originating in slow-start H.323 that require interworking to SIP, the Oracle Enterprise Session Border Controller also takes addition steps for media release in hairpinned flows that the Oracle Enterprise Session Border Controller routes as SIP to fast-start H.323.

For such a call, after the Oracle Enterprise Session Border Controller has established logical channels on the slow-start H.323 side of the call, it sends a reINVITE on the SIP side. This reINVITE has an updated session description to correct the media connection information. The the Oracle Enterprise Session Border Controller performs the hold-and-resume procedure on the fast-start side of the call. This procedure re-establishes the logical channels between the Oracle Enterprise Session Border Controller and the called endpoint, avoiding the one-way media problem.

When you are configuring H.323 globally on your Oracle Enterprise Session Border Controller , you might choose to set the noReInvite option. This option stops the Oracle Enterprise Session Border Controller from sending a reINVITE after the logical channels are established on the slow-start H.323 side of the call. Instead, the Oracle Enterprise Session Border Controller 's H.323 task communicates internally with its own SIP task a SIP UPDATE message that corrects the SDP; then the SIP task updates media flow destinations. But the Oracle Enterprise Session Border Controller does not send the UPDATE to the next hop, which can result in the one-way media problem if the call is hairpinned and media release occurs. For such cases, the default behavior for the noReInvite option is overridden. When the Oracle Enterprise Session Border Controller detects media release in an H.323-SIP call, it forwards the UPDATE to the next hop even when you enable the noReInvite option.

Dependencies

This feature depends on:

- The H.323 endpoint supports the third-party-initiated pause and re-routing feature.
- The H.323 endpoint does not change its Network Address and TSAP identifier when it re-opens the logical channels.
- The H.323 endpoint does not immediately tear down the call when there is not established logical channel in the call.
- The fact that the SIP endpoint supports the UPDATE message if the noReInvite option is enabled.

Before You Configure

The Oracle Enterprise Session Border Controller 's IWF requires that there be complete configurations for both SIP and for H.323. These two sets of configurations function together when the interworking is configured and enabled.

You enable the Oracle Enterprise Session Border Controller 's interworking capability when you set the IWF configuration's state parameter to enabled, and all required H.323 and SIP configurations are established. This means that all of the following configurations must be established:

- A full SIP configuration, including SIP interfaces, SIP ports, SIP-NATs (if needed), and SIP features
- A full H.323 configuration, including H.323 global and H.323 interface configurations
- Local policy and local policy attributes (the IWF will not work without these configurations)
- Media profiles
- Session agents and, if needed, session agent groups

H.323 Configuration

You must have a complete configuration to support H.323 traffic on your Oracle Enterprise Session Border Controller , including any required support for H.323 Fast Start or Slow Start.

In the H.323 interface configuration, you are able to configure interfaces that enable communication between H.323 devices (for audio, video, and/or data conferencing sessions).

If you know that your Oracle Enterprise Session Border Controller will be handling traffic originating in Slow Start H.323, you must establish the appropriate list of media profiles in the IWF configuration. Handling Slow Start traffic also requires that you establish appropriate local policy (and local policy attribute) configurations, but configuring session agents and session agent groups is optional.

SIP Configuration

SIP functionality must also be configured on your Oracle Enterprise Session Border Controller that will perform IWF translations. You must use appropriate local policy (and local policy attribute) configurations, but configuring session agents and session agent groups is optional. If you use session agents, then you must also configure the information you need for media profiles.

The Role of Local Policy

You must configure local policies (and local policy attributes, if necessary) in order for translations between SIP and H.323 to take place. These local policies determine what protocol is used on the egress side of a session. Local policy and local policy attribute configurations make routing decisions for the session that are based on the next hop parameter that you set. The next hop can be any of the following:

- IPv4 address of a specific endpoint
- Hostname or IPv4 address of a session agent
- Name of a session agent group

You can use the application protocol parameter in the local policy attributes configuration as a way to signal the Oracle Enterprise Session Border Controller to interwork the protocol of an ingress message into a different protocol as it makes its way to its egress destination (or next hop).

For example, if you set the application protocol parameter to SIP, then an inbound H.323 message will be interworked to SIP as it is sent to the next hop. An inbound SIP message would travel to the next hop unaffected. Likewise, if you set the application protocol parameter to H.323, then an incoming SIP message will be interworked to H.323 before the Oracle Enterprise Session Border Controller forwards it to the next hop destination.

The following example shows a configured local policy and its attributes used for IWF traffic.

```
local-policy
        from-address                            *
        to-address                              444
        source-realm                            *
        state                                   enabled
        last-modified-date                      2004-04-20 17:43:13
        policy-attribute
                next-hop                                sag:sag_internal
                realm                                   internal
                replace-uri                             disabled
                carrier
                start-time                              0000
                end-time                                2400
                days-of-week                            U-S
                cost                                    0
                app-protocol                            SIP
                state                                   enabled
                media-profiles
```

Local Policy in an IWF Session Initiated with H.323

In a session where the Oracle Enterprise Session Border Controller is interworking H.323 to SIP, it internally forwards the session on for interworking when:

- The next hop in the local policy is configured as a SIP session agent
- The next hop in the local policy is configured as a SIP session agent group

- The next hop in the local policy is not configured as a session agent or session agent group, and the application protocol parameter is set to SIP in the local policy attributes configuration.

Local Policy in an IWF Session Initiated with SIP

In a session where the Oracle Enterprise Session Border Controller is interworking SIP to H.323, it internally forwards the session on for interworking when:

- The next hop in the local policy is configured as an H.323 session agent
- The next hop in the local policy is configured as an H.323 session agent group
- The next hop in the local policy is not configured as a session agent or session agent group, and the application protocol parameter is set to H.323 in the local policy attributes configuration

  In this case the local policy should also define the egress realm, which you can set in the realm parameter of the local policy attributes configuration.

Configuring Interworking

If you have already completed the steps outlined in this chapter's IWF Service Enhancements section, then enabling the IWF is a simple process. This section shows you how to enable the IWF, and how to enable certain features that you can use to supplement basic IWF functionality.

An IWF configuration might appear like this in the ACLI:

```
iwf-config
        state                           enabled
        media-profiles
                                        PCMU
                                        telephone-event
        logging                         disabled
```

IWF Configuration

To enable the IWF on your Oracle Enterprise Session Border Controller :

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type iwf-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# iwf-config
   ```

   From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a ? at the system prompt.

4. state—Enable this parameter if you want to translate SIP and H.323 sessions on your Oracle Enterprise Session Border Controller . The default value is disabled. Valid values are:

   - enabled | disabled

5. media-profiles—Enter the name of the media profiles you want to use for IWF translations. This name is either the name of an SDP codec (such as PCMU), or it can be telephone-event if you are configuring your system for DTMF support.

   If you want to use more than one media profile for SIP/H.323 translations, enter the names in quotation marks with a space between each one.

   ```
   ACMEPACKET(iwf-config)# media-profiles "PCMU telephone-event"
   ```

6. logging—Enable this parameter if you want the Oracle Enterprise Session Border Controller to log SIP messages that are related to the IWF. The default value is disabled. Valid values are:

- enabled | disabled

Topology Hiding for IWF with an Internal Home-Realm

You can configure the Oracle Enterprise Session Border Controller to mask the IP address of the originating caller in the SIP From and/or P-Asserted-Identity headers when calls are placed from H.323 to SIP endpoints.

The option NoPAssertedID checks for incoming SETUP messages have the presentation indicator set to restricted and instructs the Oracle Enterprise Session Border Controller to send a Privacy header without the P-Asserted-Identity and not to make the From header anonymous.

The option replace-call-origin-ip removes the calling party's IP address in the SIP From header. The IP address from the internal home realm is used instead.

The topology hiding feature uses the presentation indicator field from an inbound H.323 setup message to determine if/how the headers will be masked. The following table summarizes the configurable Oracle Enterprise Session Border Controller parameters and the values for the From and P-asserted-identity headers.

H.255 Presentation Indicator Setting	P-Asserted-Identity Header Value	From Header Value	SD Session Agent Option
Allow	IP address from home realm of SD SIP Config	H.255 Calling Party IP/Port	No Option Set
Allow	IP address from home realm of SD SIP Config	IP address of Home Realm SIP-Interface	NoPAssertedID
Allow	IP address from home realm of SD SIP Config	IP address of Home Realm SIP-Interface	replace-call-origin-ip
Allow	IP address from home realm of SD SIP Config	IP address of Home Realm SIP-Interface	replace-call-origin-ip, NoPAssertedID
Restricted	PAI Header not present	Anonymous	No Option Set
Restricted	PAI Header not present	Anonymous	NoPAssertedID
Restricted	PAI Header not present	Anonymous	replace-call-origin-ip
Restricted	PAI Header not present	Anonymous	NoPAssertedID, replace-call-origin-ip

IWF Topology Hiding Configuration

To enable IWF topology hiding on your Oracle Enterprise Session Border Controller :

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Use the ACLI select command so that you can work with the session agent configuration to which you want to add these options.

```
ACMEPACKET(session-agent)# select
```

5. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (replace-call-origin-ip), and then press Enter. Follow the same steps to add the NoPAssertedID option.

```
ACMEPACKET(session-agent)# options +replace-call-origin-ip
ACMEPACKET(session-agent)# options +NoPAssertedID
```

If you type either of these options without the plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign as shown in the example above.

DTMF Support

For calls that require the IWF, you can enable support for the relay of RFC 2833 DTMF digits. The availability of this feature means that the Oracle Enterprise Session Border Controller is compliant with RFC 2833, which defines two payload formats for DTMF digits. To learn more about this RFC, refer to *http://www.ietf.org/rfc/rfc2833.txt*.

Until the exchange of TCS messages with the H.323 endpoint, the Oracle Enterprise Session Border Controller has no information about the endpoint's RFC 2833 capabilities. The Oracle Enterprise Session Border Controller adds telephone-event to the SDP on the SIP side of the call.

For calls that require SIP/H.323 translation, you can enable support for the relay of RFC 2833 DTMF digits.

To use this feature, you need to configure a media profile called telephone-event and set relevant parameters for it. Application of the media profile can happen either in a session agent configuration or in the IWF configuration.

- The name parameter in the media profiles configuration
- The media-profiles list in the IWF configuration
- The media-profiles list in the session agent configuration

All of the scenarios outlined here assume that you have established a telephone-event media profile configuration.

You can configure DTMF support using the following parameters. The way that the Oracle Enterprise Session Border Controller uses these values is described below. The payload type, part of the media profiles configuration, is dynamic and varies with different endpoints, so there is no default configuration for the telephone-event media profile.

The telephone-event media profile is used as follows in these types of IWF sessions:

- Calls that require the IWF originating in H.323 Slow Start—There is no channel (media) information available on the H.323 side.
  - If the incoming H.323 endpoint is configured as a session agent on the Oracle Enterprise Session Border Controller , then the telephone-event parameter in the media profiles set for that session agent configuration will be used in the SDP on the SIP side of the session.
  - If the H.323 endpoint is not a session agent or the telephone-event media profile is not configured in the session agent configuration corresponding to the endpoint, then the Oracle Enterprise Session Border Controller refers to the media profile information configured for the IWF configuration.
- Calls that require the IWF originating in SIP—If the TCS was not exchanged before a 200 OK was sent on the SIP side, the Oracle Enterprise Session Border Controller will behave in one of these two ways.
  - If the outbound H.323 endpoint is configured as a session agent, then the media profiles from that session agent configuration will be used.
  - If the outbound H.323 endpoint is not configured as a session agent, the media profile configured within the IWF configuration with the telephone-event value will be used.

As mentioned above, DTMF support is configured by using a combination of the telephone-event media profile and either the session agent or IWF configuration. First you set up the media profile, then you apply it to a session agent or to the IWF configuration.

DTMF Configuration

DTMF support requires you to configure a media profile named telephone-event. This section shows you how to set it up.

To configure a telephone-event media profile:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type media-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# media-profile
   ```

   From this point, you can configure parameters for media profiles. To view see all media profile configuration parameters, enter a ? at the system prompt.

4. name—Enter the name telephone-event and press Enter.
5. parameters—Enter the parameters to be applied for the codec; these are the digits that endpoints can support.
6. media-type—Leave the default media type set to audio.
7. payload-type—Set the payload type to 101, which is the dynamic payload type needed to support this feature.
8. transport—Leave the default transport protocol set to RTP/AVP.
9. frames-per-packet—You can leave this parameter set to 0 (default).
10. req-bandwidth—You can leave this parameter set to 0 (default).

Applying the Media Profile

After you have configured the telephone-event media profile, you need to apply it either to a H.323 session agent or the global IWF configuration.

DTMF for all IWF translations

To use DTMF for all IWF translations:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type iwf-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# iwf-config
   ```

   From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a ? at the system prompt.

4. Add the telephone-event media profile to the media profiles list and save your work. If you already have a media profiles for the IWF configuration set up and want to keep them (adding telephone-event to the list), then you must type in all of the media profiles that you want to use.

   ```
   ACMEPACKET(iwf-config)# media-profiles "PCMU telephone-event"
   ACMEPACKET(iwf-config)# done
   ```

DMTF Support on a Per-Session-Agent Basis

To use DMTF support on a per-session-agent basis:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
```

From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a ? at the system prompt.

4. When you configure a new H.323 session agent, you can configure DTMF support by simply adding the telephone-event media profile to the list of media profiles. You can add it along with the other media profiles you might want to use for that session agent.

```
ACMEPACKET(session-agent)# media-profiles "telephone-event g711Ulaw64k"
```

5. When you want to add DTMF support to an H.323 session agent that you have already configured, you need to select that session agent, add the media profile, and save your work.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: 192.168.1.48 realm=         ip=
2: 192.168.1.49 realm=         ip=
3: 192.168.1.50 realm=external ip=
selection:3
ACMEPACKET(session-agent)# media-profiles "telephone-event g711Ulaw64k"
ACMEPACKET(session-agent)# done
```

RFC 2833 DTMF Interworking

This section explains the Oracle Enterprise Session Border Controller's support of transporting Dual Tone Multi-Frequency (DTMF) in Real-Time Transport Protocol (RTP) packets (as described in RFC 2833) to H.245 User Input Indication (UII) or SIP INFO method interworking.

Multimedia devices and applications must exchange user-input DTMF information end-to-end over IP networks. The Oracle Enterprise Session Border Controller provides the interworking capabilities required to interconnect networks that use different signaling protocols. Also, the Oracle Enterprise Session Border Controller provides DTMF translation to communicate DTMF across network boundaries.

The Oracle Enterprise Session Border Controller supports:

- RFC 2833 to H.245 UII translation for H.323-to-H.323 calls, when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device that only uses H.245 UII.
- RFC 2833 to H.245 UII or INFO translation of H.323 to SIP (and SIP to H.323) IWF calls, when one side is a version 4 H.323 device requiring RFC 2833 DTMF event packets and the SIP endpoint only supports INFO messages. Or when one side is a pre-version 4 H.323 device that only uses H.245 UII and the SIP endpoint supports RFC-2833 DTMF event packets.

About RFC 2833

RFC 2833 specifies a way of encoding DTMF signaling in RTP streams. It does not encode the audio of the tone itself, instead a signal indicates the tone is being sent. RFC 2833 defines how to carry DTMF events in RTP packets. It defines a payload format for carrying DTMF digits used when a gateway detects DTMF on the incoming messages and sends the RTP payload instead of regular audio packets.

About H.245 UII

H.245 provides a capability exchange functionality to allow the negotiation of capabilities and to identify a set of features common to both endpoints. The media and data flows are organized in logical channels. H.245 provides logical channel signaling to allow logical channel open/close and parameter exchange operations. The H.245 signaling protocol is reliable, which ensures that the DTMF tones will be delivered.

H.245 User Input Indication (UII) plays a key role in all the services that require user interaction. For video messaging, typical uses of UII include selection of user preferences, message recording and retrieval, and typical mailbox management functions. H.245 UII provides two levels of UII, alphanumeric and signal.

About RFC 2833 to H.245 UII Interworking

The Oracle Enterprise Session Border Controller provides 2833 to H.245-UII interworking by checking 2833-enabled RTP streams for packets matching the payload type number for 2833. It then sends the captured packet to the host for processing and translation to H.245 UII messages. A H.245 UII message received by the Oracle Enterprise Session Border Controller is translated to 2833 packets and inserted into the appropriate RTP stream.

About DTMF Transfer

DTMF transfer is the communication of DTMF across network boundaries. It is widely used in applications such as interactive voice response (IVR) and calling card applications.

The multiple ways to convey DTMF information for packet-based communications include:

- In-band audio: DTMF digit waveforms are encoded the same as voice packets. This method is unreliable for compressed codecs such as G.729 and G.723
- Out-of-band signaling events:

  H.245 defines out-of-band signaling events (UII) for transmitting DTMF information. The H.245 signal or H.245 alphanumeric methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of through the RTP channel. The tones are transported in H.245 UII messages.

  All H.323 version 2 compliant systems are required to support the H.245 alphanumeric method, while support of the H.245 signal method is optional.

  SIP INFO – uses the SIP INFO method to generate DTMF tones on the telephony call leg. The SIP INFO message is sent along the signaling path of the call. Upon receipt of a SIP INFO message with DTMF content, the gateway generates the specified DTMF tone on the telephony end of the call.
- RTP named telephony events (NTE): uses NTE to relay DTMF tones, which provides a standardized means of transporting DTMF tones in RTP packets according to section 3 of RFC 2833.

Of the three RTP payload formats available, the Oracle Enterprise Session Border Controller supports RTP NTE. NTE is most widely used for SIP devices but is also supported in H.323 version 4 or higher endpoints.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for 2833 telephone event packets using a SDP or H.245 Terminal Capability Set (TCS), which tells the other side what payload type number to use when sending the named event packets to the local device. Most devices use payload type number 101 for 2833 packets, although no default is specified in the standard.

The 2833 packet's RTP header also makes use of the timestamp field. Because events often last longer than the 2833 packets sending interval, the timestamp of the first 2833 packet an event represents the beginning reference time for subsequent 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

See RFC 2833 and draft-ietf-avt-rfc2833bis-07.txt for more information.

Preferred and Transparent 2833

To support preferred (signaled) 2833 and transparent 2833, the Oracle Enterprise Session Border Controller provides 2833 detection and generation (if necessary) when the endpoint signals support for 2833.

- Preferred: the Oracle Enterprise Session Border Controller only generates and detects 2833 for endpoints if they negotiate support for 2833 through signaling

- Transparent: the Oracle Enterprise Session Border Controller offers and answers based on end-to-end signaling and transparently relaying 2833

Preferred 2883 Support

If one side of the call, or a SIP interface, or a session agent, is configured for preferred 2833, the Oracle Enterprise Session Border Controller only generates and detects 2833 for endpoints if they signal support for 2833. The Oracle Enterprise Session Border Controller will offer 2833 in the TCS SDP, even if the originating caller did not.

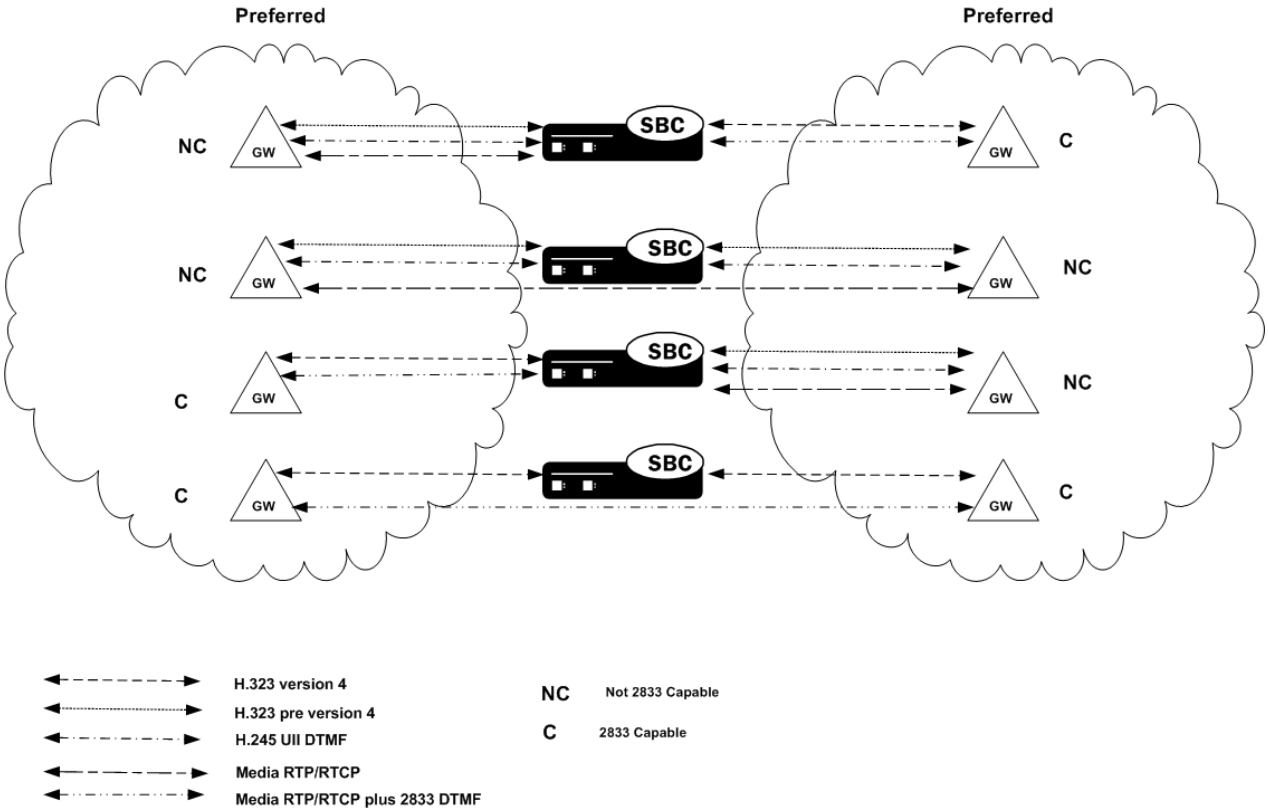
- When the Oracle Enterprise Session Border Controller manages calls originating from a preferred source going to a preferred target, it:

  Performs 2833 translation for an endpoint when the originating side requests 2833 but the target does not negotiate 2833

  Allows 2833 to pass through if the originating side and target of the call are configured as preferred and negotiate 2833
- When the Oracle Enterprise Session Border Controller manages calls originating from a preferred source going to a transparent target, it:

  Performs 2833 translation when the originating side requests 2833 but the target is configured as transparent and does not negotiate 2833.

  Allows 2833 to pass through if the originating side and the target of the call are configured as transparent and negotiate 2833. The Oracle Enterprise Session Border Controller does not perform active translation because both ends support 2833.



If one SIP endpoint does not signal 2833 capability, but the other SIP or H.323 endpoints do, the Oracle Enterprise Session Border Controller does not perform 2833 translation.

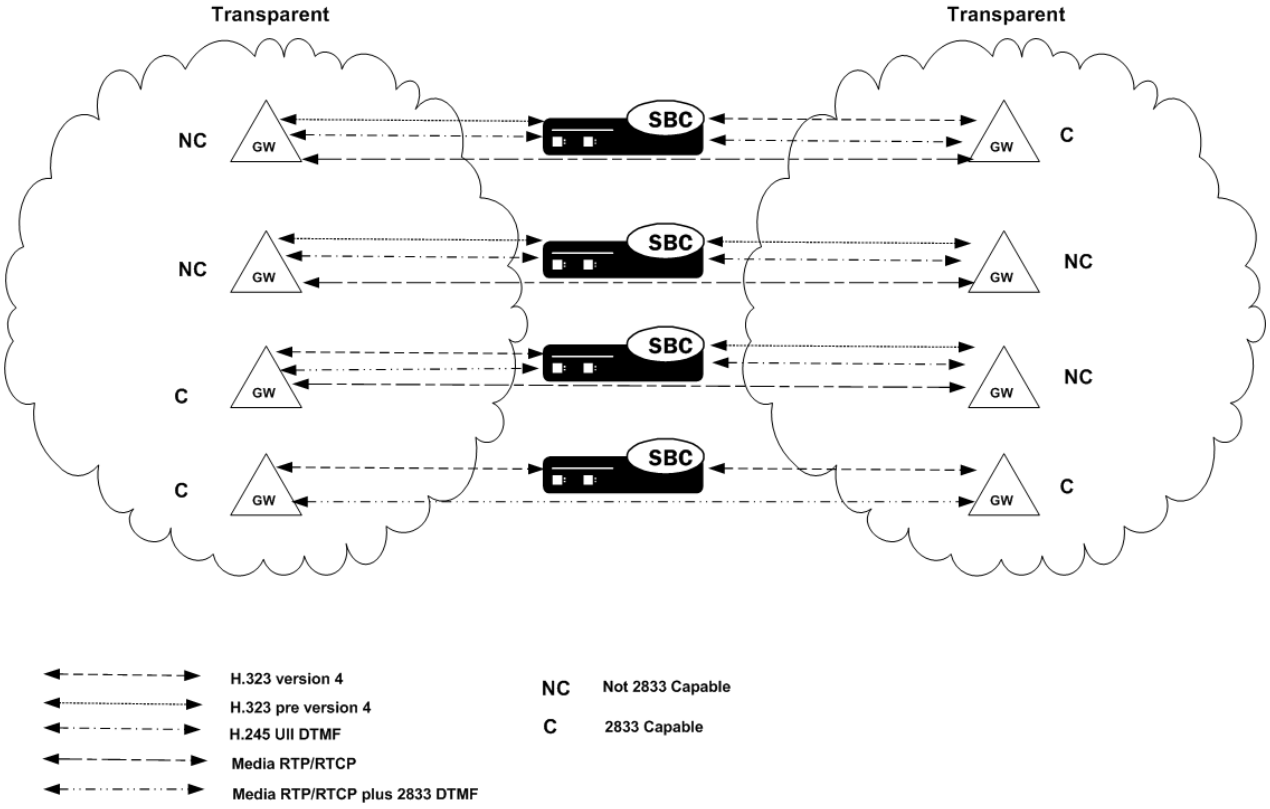
Transparent 2833 Support

The default configuration of the Oracle Enterprise Session Border Controller for H.323 is transparent 2833. The Oracle Enterprise Session Border Controller passes on the offered capabilities to the next-hop signaling element. If

the next-hop endpoint is for a transparent 2833 target, typical capability negotiation determines the DTMF method. The Oracle Enterprise Session Border Controller transparently relays the DTMF as it has in previous releases.

With transparent 2833, the Oracle Enterprise Session Border Controller acts as a typical B2BUA or B2BGW/GK. However when the target of the call is configured as preferred 2833, the Oracle Enterprise Session Border Controller:

- Relays the 2833 packets if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target signals 2833
- Performs 2833 translation if the originating endpoint does not signal 2833 and the next-hop endpoint for the preferred target does signal 2833
- Does not perform 2833 translation or transparently relay 2833 if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target (or even a transparent 2833 target) does not signal 2833.



Payload Type Handling

The Oracle Enterprise Session Border Controller supports the RTP NTE for telephony events such as transport of DTMF digits and hook flash. Using RTP NTE, endpoints perform per-call negotiation of the DTMF transfer method and negotiate payload type value for the RTP NTE packets.

Although most endpoints use payload type number 101, the RTP payload type formats can become asymmetrical when being interworked between SIP and H.323 because there is no default standard and endpoints use different types. This means that the payload type negotiated on one side of the Oracle Enterprise Session Border Controller, and that ends up being used for the call, might not be the same payload type negotiated on the other side of the Oracle Enterprise Session Border Controller. And while certain endpoints handle the asymmetry well, others do not.

Consider the simplified example of an IWF call initiated in SIP and translated to H.323. In this scenario, the SIP endpoint negotiates the payload type 106 with the Oracle Enterprise Session Border Controller. And despite the fact that the H.323 endpoint negotiates payload type 101, the Oracle Enterprise Session Border Controller returns type 106 and the call proceeds using type 106.

However, you can enable forced symmetric payload type handling so the Oracle Enterprise Session Border Controller changes the payload type of RFC 2833 packets to avoid using asymmetrical payload types.

For H.323 session agents and H.323 interfaces (stacks), you can configure an option that forces symmetric payload type use. The Oracle Enterprise Session Border Controller can detect when the payload types negotiated by the SIP and H.323 endpoints are symmetrical and when they are not. When it detects asymmetrical payload type use, the Oracle Enterprise Session Border Controller forces the remote endpoint to use the RFC 2833 payload type you configure in the SIP interface.

Basic RFC 2833 Negotiation Support

If H.323, SIP, or session agents on either side of the call are configured for preferred 2833 support, the Oracle Enterprise Session Border Controller supports end-to-end signaled negotiation of DTMF on a call-by-call basis. If the calling party is not configured for preferred support but sends 2833, the Oracle Enterprise Session Border Controller sends 2833 to the next-hop called party. If the calling party sends H.245 signals or alphanumeric UII, the Oracle Enterprise Session Border Controller sends H.245 signals or alphanumeric UII to the next-hop called party (if it is an H.323 next-hop).

The Oracle Enterprise Session Border Controller also supports hop-by-hop negotiation of DTMF capability on a call-by-call basis, if the signaling protocols or session agents on either side of the call are configured for preferred 2833 support.

H.323 to H.323 Negotiation

The Oracle Enterprise Session Border Controller serves as the H.323 called gateway. It answers RFC 2833 audio telephony event capability in the version 4 H.323/H.245 TCS when it receives a call from an H.323 endpoint configured for preferred RFC 2833.

If the Oracle Enterprise Session Border Controller is the answering device, configured for preferred support, and the calling device sends 2833, the Oracle Enterprise Session Border Controller accepts the 2833 regardless of the next-hop's DTMF capabilities. The received dynamic RTP payload type is used for detecting 2833 packets, while the response dynamic payload type is used for generating 2833 packets.

The Oracle Enterprise Session Border Controller supports:

- RFC-2833 audio telephony events in the version 4 H.323/H.245 TCS as the H.323 calling gateway, when the Oracle Enterprise Session Border Controller calls an H.323 endpoint configured for preferred RFC 2833 support. The Oracle Enterprise Session Border Controller sends 2833 to the called party regardless of whether the calling party sends it.
- H.245 UII and RFC-2833 packets sent at the same time, to the same endpoint, even if only half of the call is being provided 2833 support by the Oracle Enterprise Session Border Controller.

  If one half of the call supports H.245 UII, and the other half is being provided 2833 translation by the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller can also forward the H.245 UII it receives to the 2833 endpoint. For example, when the signaling goes through a gatekeeper or third party call control, sending the H.245 UII in the signaling path allows those devices to learn the DTMF digits pressed.

Signal and Alpha Type Support

The Oracle Enterprise Session Border Controller supports:

- H.245 signal and alpha type UII in the H.323/H.245 TCS as the H.323 calling gateway when the:

  Oracle Enterprise Session Border Controller calls an H.323 endpoint configured for transparent 2833 support

  calling endpoint's target is configured as preferred

  If the originating preferred side also sends 2833, the Oracle Enterprise Session Border Controller forwards it to the transparent side. The Oracle Enterprise Session Border Controller sends signal and alpha UII support to the called party regardless of whether the calling party sends it, if the call originates from a preferred side to a transparent side.
- H.245 alphanumeric UII for DTMF for H.323 endpoints that do not signal 2833 or contain explicit H.245 UII capability, for stacks configured for transparent 2833 support.

When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Oracle Enterprise Session Border Controller translates incoming H.245 UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Oracle Enterprise Session Border Controller relays the H.245 UII messages.

- H.245 signal type UII for DTMF for H.323 endpoints that do not signal 2833, but do signal explicit H.245 UII capability, for stacks configured for transparent 2833 support.

  When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Oracle Enterprise Session Border Controller translates incoming H.245 signaled UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Oracle Enterprise Session Border Controller relays the H.245 UII messages if both sides support it.

H.323 to SIP Calls

This section explains DTMF interworking specific to H.323 to SIP calls.

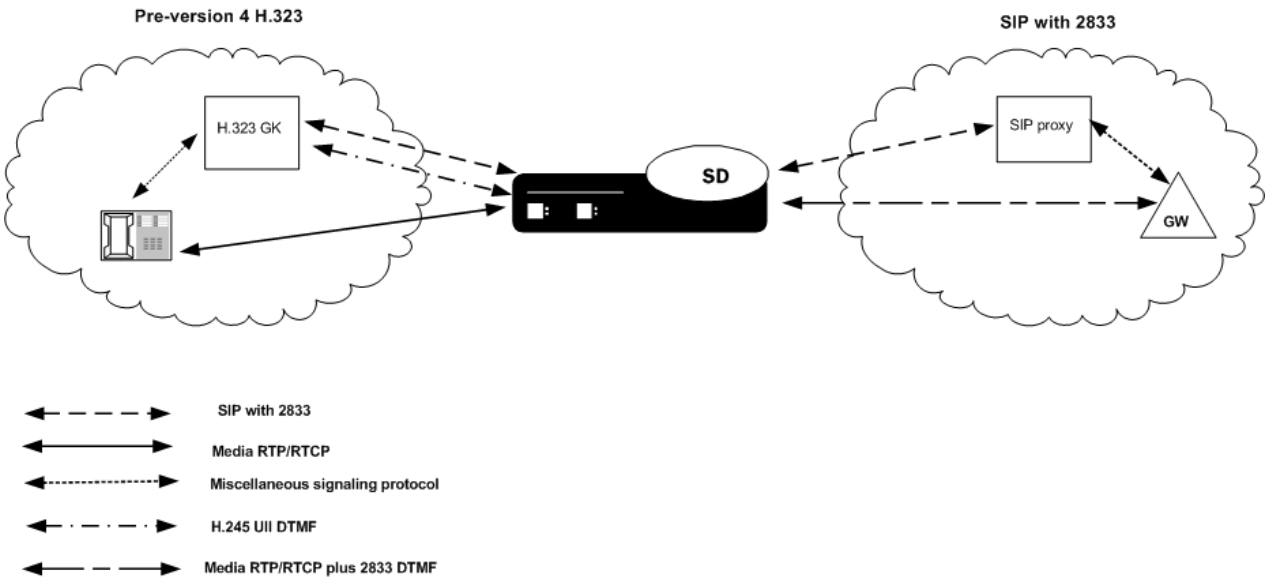
SIP Endpoints

SIP endpoints include those that support:

- RFC 2833
- SIP INFO method

H.323 Non-2833 Interworking with SIP

RFC 2833 and the SIP INFO method can be used for conveying DTMF information for SIP based-services. (RFC 2833 is the most widely used.) To provide end-to-end DTMF for SIP devices supporting RFC-2833 interworking with H.323 devices that do not, an RFC 2833 to H.323 UII interworking function is provided.



How H.323 to SIP Calls Work

For H.323 to SIP IWF calls, if 2833-related information is to be sent in the INVITE, the SIP interface of the SIP session agent has to be configured with the rfc2833-mode parameter set to preferred.

The following example shows an INVITE without 2833 in the SDP:

```
Apr  5 04:28:50.073 On 127.0.0.1:5070 sent to 127.0.0.1:5060
INVITE sip:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5070;branch=z9hG4bKIWF0000gl2018604agg71c0;acme_irealm=external;acme
_sa=192.168.1.6
```

```
Contact: "jdoe"<sip:127.0.0.1:5070>
GenericID: 114421133000000@000825010100
Supported: 100rel^M
From: "msmith"<sip:192.168.200.68:5060>;tag=000000ab00011940
To: <sip:780@192.168.200.6:5060>
Call-ID: 7f00000113ce000000ab000101d0@127.0.0.1
CSeq: 2 INVITE
Content-Length: 225
Content-Type: application/sdp
v=0
o=IWF 3 3 IN IP4 192.168.1.6
s=H323 Call
c=IN IP4 192.168.1.6
t=0 0
m=audio 5214 RTP/AVP 0  18
a=rtpmap:0 PCMU/8000/1
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
m=video 5216 RTP/AVP 31
a=rtpmap:31 H261/9000/1
```

SIP INFO—RFC 2833 Conversion

The Oracle Enterprise Session Border Controller can perform SIP INFO—RFC 2833 conversion. The Oracle Enterprise Session Border Controller also provides a way for you to enable a dual conversion mode, where the Oracle Enterprise Session Border Controller:

- Inserts telephone-event in the SDP offer
- Generates both RFC 2833 event packets and SIP INFO messages regardless of whether or not the SDP offer indicates RFC 2833

You can enable this feature either for SIP interfaces or session agents. The following apply:

- If the next hop SIP interface or session agent's rfc2833-mode is set to preferred, then the SD inserts RFC 2833 into the SDP offer/answer. This occurs regardless of whether:
  - The original SDP on the opposite side of the call does not support RFC 2833
  - The opposite side's SIP interface or session agent is set to transparent mode
- If the next hop SIP interface or session agent is set to transparent, then the behavior of the Oracle Enterprise Session Border Controller depends on the previous hop.
  - If the previous hop is a SIP interface or session agent configured for transparent mode, then the S Oracle Enterprise Session Border Controllerdoes not perform any conversion.
  - If the previous hop is a SIP interface or session agent configured for preferred mode, the Oracle Enterprise Session Border Controller does not insert RFC-2833 into the SDP on the transparent side. It does, however, translate from RFC 2833 to SIP INFO if the originating endpoint supports RFC 2833.

IPv6 SIP INFO to RFC 2833 Telephone Event Interworking

The Oracle Enterprise Session Border Controller can interwork SIP INFO and RFC Telephone Event messages for IPv4, IPv6—or for any session requiring interworking between IPv4 and IPv6. Other than installing the applicable lisences on your Oracle Enterprise Session Border Controller and enabling IPv6 support in your system configuration (system-config), you do not have to perform any configuration steps for this capability to work.

RFC 2833 Interworking Configuration

This section explains how to configure RFC 2833 to H.245 User Input Indication (UII) or SIP INFO method interworking.

RFC 2833 Mode for H.323 Stacks

To configure RFC 2833 mode for H.323 stacks:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

   From this point, you can configure H.323 stack parameters. To view all H.323 stack parameters, enter a ? at the system prompt.

5. rfc2833-mode—Set the RFC2833 mode. The default value is transparent. Valid values are:

   - transparent—The Oracle Enterprise Session Border Controller and H.323 stack behave exactly the same way as before and the 2833 or UII negotiation is transparent to the Oracle Enterprise Session Border Controller.
   - preferred—The H.323 stack uses 2833 for DTMF transfer, which it signals in its TCS. However, the remote H323 endpoint makes the decision. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack reverts back to using UII. You configure the payload format by configuring the h323-config element.

RFC 2833 Payload for H.323

To configure the RFC 2833 payload in preferred mode:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323
```

   From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt.

4. rfc2833-payload—Enter a number that indicates the payload type the Oracle Enterprise Session Border Controller will use for RFC 2833 packets while interworking 2833 and UII. The default value is 101.

   - Minimum—96
   - Maximum—127

Configuring the SIP Interface

You configure the 2833 mode and payload for the SIP interface. You must configure the payload the Oracle Enterprise Session Border Controllerwill use for RFC 2833 packets, while interworking 2833 and INFO/UII.

To configure the SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.

4. rfc2833-payload—Enter a number that indicates the payload type the SIP interface will use for RFC 2833 packets while interworking 2833 and UII. The default value is 101.The valid range is:
   - Minimum—96
   - Maximum—127
5. rfc2833-mode—Set the RFC 2833 mode for the SIP interface. The default value is transparent. Valid values are:
   - transparent—The SIP INFO and RFC 2833 translation is transparent to the Oracle Enterprise Session Border Controller.
   - preferred—The RFC 2833 transfer method is the preferred method for sending DTMF, and a telephone event is inserted in the SDP of the outgoing offer. The actual method of transfer, however, depends on the SDP offer/answer exchange that occurs between the Oracle Enterprise Session Border Controller and remote endpoint. If the remote endpoint supports RFC 2833, the Oracle Enterprise Session Border Controller performs SIP INFO —RFC 2833 conversion.
   - dual—The Oracle Enterprise Session Border Controller behaves the same as it does when set to preferred mode, and it forwards both the original DTMF mechanism and the translated one to the remote endpoint.

Configuring Session Agents

You configure session agents with:

- payload type the Oracle Enterprise Session Border Controller wants to use for RFC 2833 packets while interworking 2833 and UII.

  The default value for this attribute is 0. When this value is zero, the global rfc2833-payload configured in the h323-configuration element will be used instead. For SIP session agents, the payload defined in the SIP interface is used, if the SIP interface is configured with the preferred RFC 2833 mode.
- 2833 mode

  A value of transparent or preferred for the session agent's 2833 mode will override any configuration in the h323-stack configuration element.

  To configure session agents:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```
3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```
4. rfc2833-mode—Set the RFC 2833 mode you want the session agent to use. The default value is none. Valid values are:
   - none—RFC 2833 to UII interworking is based on the H.323 stack configuration.
   - transparent—The RFC 2833 or UII negotiation is transparent to the Oracle Enterprise Session Border Controller. This overrides the H.323 stack configuration, even if the stack is configured for preferred mode.
   - preferred—RFC 2833 for DTMF transfer is preferred, which is signaled in the TCS. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack configured as preferred will revert back to using UII. This overrides any configuration in the h323-stack even if the stack is configured for transparent mode.

     For SIP INFO—RFC 2833 conversion, you can choose:
   - none—The 2833-SIP INFO interworking will be decided based on the sip-interface configuration.

- transparent—The session agent behaves the same as it did without the SIP INFO—RFC 2833 conversion feature. The SIP INFO and RFC 2833 translation is transparent to the Oracle Enterprise Session Border Controller.
- preferred—The RFC 2833 transfer method is the preferred method for sending DTMF, and a telephone event is inserted in the SDP of the outgoing offer. The actual method of transfer, however, depends on the SDP offer/answer exchange that occurs between the Oracle Enterprise Session Border Controller and remote endpoint. If the remote endpoint supports RFC 2833, the Oracle Enterprise Session Border Controller performs SIP INFO—RFC 2833 conversion.
- dual—The Oracle Enterprise Session Border Controller behaves the same as it does when set to preferred mode, and it forwards both the original DTMF mechanism and the translated one to the remote endpoint.

5. rfc2833-payload—Enter a number that indicates the payload type the session agent will use for RFC 2833 packets while interworking 2833 and UII. The default value is 0. The valid range is:

   - Minimum—0, 96
   - Maximum—127

Enabling Payload Type Handling

You can configure H.323 session agents and H.323 interfaces (stacks) with an option that forces symmetric payload type use. For Payload Type Handling to work properly, you must set the following SIP interface and the global H.323 configuration parameters with these values:

- rfc2833-mode—Set this parameter to preferred; the default is transparent.
- rfc2833-payload—Set this parameter to the payload type you want forced for the remote endpoint. Your entry will be between 96 and 127, with 101 as the default.

To enable forced symmetric payload type handling for an H.323 session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you want to add this option to a pre-existing H.323 session agent, select the one you want to edit.

4. options—Set the options parameter by typing options, a Space, the option name Map2833ForceRemotePT with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(session-agent)# options +Map2833ForceRemotePT
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

To enable forced symmetric payload type handling for an H.323 interface:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type h323-config and press Enter.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

9. Type h323-stacks and press Enter.

```
ACMEPACKET(h323-config)# h323-stacks
ACMEPACKET(h323-stack)#
```

10. options—Set the options parameter by typing options, a Space, the option name Map2833ForceRemotePT with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(h323-stack)# options +Map2833ForceRemotePT
```

    If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

11. Save and activate your configuration.

DTMF Transparency for IWF

In certain vendors' implementations of DTMF during SIP/H.323 IWF, there have been discrepancies between the RFC 2833 and UII/INFO negotiations and what type of messages actually get sent. Instead of correcting these errors on its own end, the Oracle Enterprise Session Border Controller has perpetuated these inaccuracies.

To ensure that the Oracle Enterprise Session Border Controller always sends the correctly negotiated protocols, a media-manager-config parameter called translate-non-rfc2833-event has been created. When translate-non-rfc2833-event is enabled, the Oracle Enterprise Session Border Controller always sends the type of messages that were initially negotiated, regardless of the type of messages it may be receiving.

DTMF Transparency Configuration

To enable DTMF transparency for SIP/H.323 IWF:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type media-manager and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. translate-non-rfc2833-event—To enable this feature, set this parameter to enabled. If you do not want to use the feature leave it set to its default behavior, disabled.
5. Save and activate your configuration.

RFC 2833 Packet Sequencing

You can configure your Oracle Enterprise Session Border Controller to generate either the entire start-interim-end RFC 2833 packet sequence or only the last three end 2833 packets for non-signaled digit events.

RFC 2833 Packet Sequencing Configuration

To send only the last three end 2833 packets for non-signaled digits events:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type media-profile and press Enter.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. rfc2833-end-pkts-only-for-non-sig—By default, this parameter is enabled—meaning that only the last three end 2833 packets are used for non-signaled digits events. Set this parameter to disabled if you want the entire start-interim-end RFC 2833 packet sequence for non-signaled digit events
5. Save and activate your configuration.

Enhanced H.245 to 2833 DTMF Interworking

Enhanced H.245 to 2833 and SIP INFO to 2833 DTMF interworking addresses issues experienced where the way the Oracle Enterprise Session Border Controller timestamps audio RTP packets result in dropped digits and digits with a stutter pattern. These occurrences can cause other network devices to deem the packets unrecoverable (due to jitter), meaning that they will never render the digit.

The Oracle Enterprise Session Border Controller offers the following:

- Timestamp is based on the current time—The Oracle Enterprise Session Border Controller can compute the timestamp of the egress 2833 packets using the actual time elapsed in milliseconds since the last RTP packet (rather than incrementing the time by 1 sample). Not only does the Oracle Enterprise Session Border Controller fill out the timestamp field more accurately, but it also recalculates the checksum.
- End-event 2833 messages default behavior—The Oracle Enterprise Session Border Controller's new default behavior is to send three end-event 2833 packets only if the DTMF event is received for:
  - An alphanumeric UII or SIP INFO with no duration
  - A signaled UII or SIP INFO with a duration less than the minimum signal duration (the value you configure using the new media manager configuration min-signal-duration option)

  For a signaled UII or SIP INFO with a duration greater than the minimum signal duration, the Oracle Enterprise Session Border Controller behaves as it does in prior releases: It sends the initial event packets, any interim packets (if they exist), and the three end packets.
- Configurable duration for the 2833 event—Without the enhancements being configured, the Oracle Enterprise Session Border Controller uses a 250 millisecond duration for the 2833 event when it receives an alphanumeric UII or a SIP INFO with no specified duration. The result is that 2833 packets are sent at 50-millisecond intervals until the 250 millisecond time expires; then the three end-event packets are sent.

  Now the Oracle Enterprise Session Border Controller allows you to set the duration of these 2833 events using a new default-2833-duration parameter (with a 100 millisecond default) in the media manager configuration. In addition, the Oracle Enterprise Session Border Controller uses this configured value (instead of the duration sent in the signaling message) when it receives an UII or SIP INFO with a duration less than the minimum signal duration. It checks to make sure that the value for the default-2833-duration parameter is greater than the minimum signal duration.
- Configurable minimum signal duration value—Without this configured, the Oracle Enterprise Session Border Controller accepts and uses the duration it receives in the UII or SIP INFO for the 2833 event. However, you can configure this value using the min-signal-duration option in the media manager configuration. If the duration the Oracle Enterprise Session Border Controller receives is less than the threshold, it uses the value configured in the default-2833-duration parameter.

  If you do not configure this option, then there is no signaling duration threshold.

  Note: Timestamp changes and duration changes only take effect when the 2833 timestamp (rfc-2833-timestamp) is enabled in the media manager configuration.

Enhancements Configuration

This section shows you how to configure enhancements for H.245 UII/SIP INFO—2833 DTMF interworking.

To enable the Oracle Enterprise Session Border Controller to calculate the timestamp based on the current time:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type media-profile and press Enter.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. rfc-2833-timestamp—Enable this parameter to use a timestamp value calculated using the actual time elapsed since the last RTP packet. The default is disabled. Valid values are:

   - enabled | disabled

5. Save and activate your configuration.

   To configure a duration for the 2833 event:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

8. Type media-profile and press Enter.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

9. default-2833-duration—Set this parameter to the time value in milliseconds for the Oracle Enterprise Session Border Controller to use when it receives an alphanumeric UII or a SIP INFO with no specified duration; then the three end-event packets are sent. The default value is 100. The valid range is:

   - Minimum—50
   - Maximum—5000

10. Save and activate your configuration.

Setting the Minimum Signal Duration

To configure the minimum signal duration value:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type media-profile and press Enter.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. options—Set the options parameter by typing options, a Space, the option name min-signal-duration=x (where x is the value in milliseconds you want to use for the threshold) with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(media-manager-config)# options +min-signal-duration=200
```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to the configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

SIP Tel URI Support

The Oracle Enterprise Session Border Controller maps H.323 addresses to either SIP URIs or Tel URIs. You can configure the Oracle Enterprise Session Border Controller to include Tel URIs in the following SIP headers for calls that require the IWF:

- Request Line
- To
- From

When Tel URI support is not used on a Oracle Enterprise Session Border Controller performing IWF translations, the SIP INVITE is formatted like it is in the following example. This example uses 192.168.5.5 as the external proxy address, or the next hop (as configured in the local policy).

```
INVITE sip:602@192.168.5.5:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.58:5060;branch=z9hG4bKIWF0aqoqg001g11a7kos4g0
Contact: <sip:603@192.168.5.58:5060>
From: <sip:603@192.168.5.58:5060>;tag=4069ac210018a0
To: <sip:602@192.168.5.5:5060>
```

In the example above, the session needs to be routed to another SIP proxy that can resolve an E.164 number to a SIP address. However, the next SIP proxy must be informed that the message will be routed based on the included E.164 number; the SIP address of the Request URI does not have a routable SIP address. To devise a routable address, the Request URI must be reconstructed as a Tel URI.

Without Tel URI support configured, the terminating SIP user would be required to have an address of 602@192.168.5.5, where the IPv4 address portion is the same as the address for the proxy. If it were not the same, then the session would terminate at the proxy. However, the proxy would be unable to handle the session because the SIP address it received would be unknown/unroutable.

Because it is not desirable to have an IPv4 address be the user-identity and rely on the configuration of the IP network, the SIP INVITE generated by the Oracle Enterprise Session Border Controller and sent to the proxy must have the following format if it is sent to an H.323 entity.

```
INVITE tel:2345 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.52:5060;branch=z9hG4bKIWFaqoqq00cobgf9so10o0
Contact: <sip:1234@192.168.5.58:5060>
From: <tel:1234>;tag=4069ac35000c5ff8
To: <tel:2345>
Call-ID:7f0000113ce4069ac35000c5440
CSeq: 1 INVITE
Content-Length: 155
Content-Type: application/sdp
```

SIP Interface Configuration

You enable this feature in the SIP interface configuration.

To configure SIP Tel URI support for calls that require the IWF:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ```

   From this point, you can configure SIP interface parameters. To view see all SIP interface parameters, enter a ? at the system prompt.

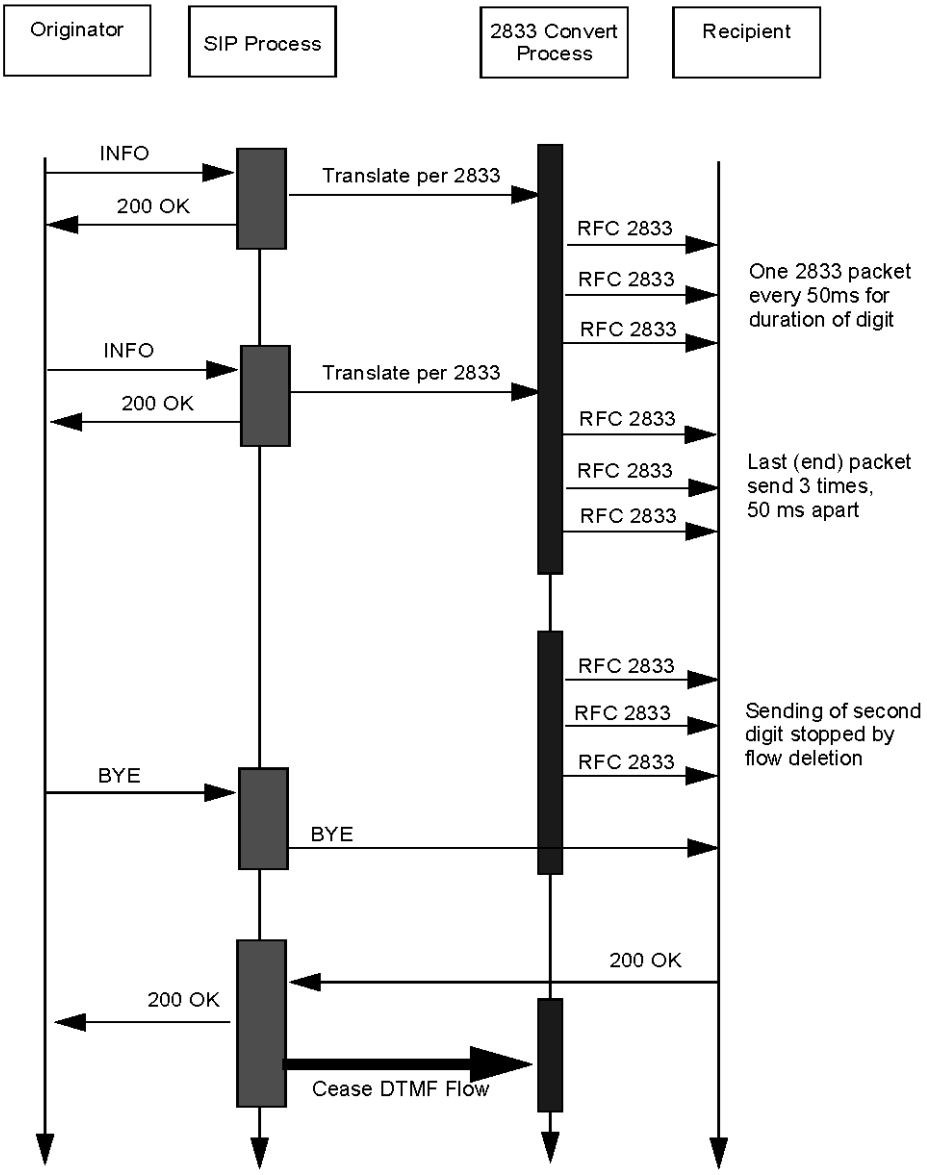
4. teluri-scheme—Enable or disable the conversion of SIP URIs to Tel URIs. The default value is disabled. Valid values are:

   - enabled | disabled

```
ACMEPACKET(sip-interface)# teluri-scheme enabled
ACMEPACKET(sip-interface)# done
```

Graceful DTMF Conversion Call Processing

The default implementation for SIP INFO to RFC2833 events performs well in most network topologies is shown below.



When the Oracle Oracle Enterprise Session Border Controller receives an INFO DTMF request, the SIP process determines whether or not it needs to perform DTMF-to-RFC2833 translation. If translation is required, the process forwards the DTMF to the 2833 convert process for translation and transmission to the recipient. Immediately after off-loading the DTMF, the SIP process sends a 200 OK response for the INFO. As shown in the figure, the 2833 convert process generates a number of RFC2833 packets to represent received DTMF digits.

Specifically, the 2833 convert process generates one RFC 2833 packet every 50 milliseconds for the duration of the DTMF digit, whose length is specified in the INFO request, and two retransmits of the last packet (known as the end packet) 50 milliseconds apart.

Consequently, the time interval between the 200 OK and the actual transmission of the RFC 2833 translation is the sum of the DTMF duration and 100 ms.

Note: This time interval can be shortened to 100 ms by enabling the rfc2833-end-pkts-only-for-non-sig parameter in media-manger which results in Oracle Enterprise Session Border Controller only generating the last packet and its two retransmits.

The early 200 OK allows the endpoint to send the next DTMF digit before the SD has sent all the RFC2833 packets, resulting in the next digit being queued internally by the 2833 convert process before being sent.

A problem arises if the SIP process receives a BYE request from the DTMF originator while queued digits are awaiting translation and transmission. In such an event, the SIP process immediately forwards the BYE request to the recipient, ending the session with DTMF digits awaiting translation and transmission.

An alternative DTMF conversion model provides for a feedback mechanism from the 2833 convert process to the SIP process. With this model enabled, the SIP process buffers a received BYE until it obtains confirmation that all queued DTMF digits have been translated and transmitted. Only after obtaining confirmation, does it forward the BYE to terminate the session.

This processing model is enable by a SIP option, sync-bye-and-2833, and requires that rfc2833-mode parameter on the egress interfaces is NOT set to dual, any value other than dual, is supported.

1. From superuser mode, use the following command sequence to access sip-config configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

2. Use the SIP option sync-bye-and-2833 to delay BYE processing until DTMF-to-RFC2833 translation has been completed.

```
ACMEPACKET(sip-config)# options +sync-bye-and-2833=”enabled”
ACMEPACKET(sip-config)#
```

3. Use the done and exit commands to complete configuration.

```
ACMEPACKET(sip-config)# done
ACMEPACKET(sip-config)# exit
ACMEPACKET(session-router)#
```

IWF Inband Tone Option

This option enables the Oracle Enterprise Session Border Controller to send a progress indicator (PI) =8 in an H.225 message when an SDP is received in a provisional message. In effect, this option sends network announcements inband. It is also applicable because in some networks H.323 endpoints support early H.245.

The H.323 inband tone option is enabled by adding the inbandTone as an option in a configured H.323 stack.

When this option is not used, the ringtone is generated locally (NO PI=8 in PROGRESS OR ALERTING) is the default behavior.

IWF Inband Tone Configuration

To configure the IWF inband tone option:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stacks)#
```

5. Use the ACLI select command add this feature to an existing H.323 interface.

```
ACMEPACKET(h323-stacks)# select
```

6. If you are adding this service to a new H.323 interface, type option inbandTone and press Enter.

   ACMEPACKET(h323-stacks)# option inbandTone

7. If you are adding this service to an H.323 interface that already exists, type select to select the interface to which you want to add the service. Then use the options command and prepend the option with a plus (+) sign.

   - If you know the same of the interface, you can type the name of the interface at the name: prompt and press Enter.
   - If you do not know the name of the interface, press Enter at the name: prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press Enter.
   - If are adding service to an existing interface and type in the option without a plus (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +inbandTone.

RFC 3326 Support

This section explains the Oracle Enterprise Session Border Controller's ability to map Q.850 cause values with SIP responses for calls that require IWF.

RFC 3326 defines a header that might be included in any in-dialogue request. This reason header includes cause values that are defined as either a SIP response code or ITU-T Q.850 cause values. You can configure the Oracle Enterprise Session Border Controller to support sending and receiving RFC 3326 in SIP messages for:

- Mapping H.323 Q.850 cause values to SIP responses with reason header and cause value
- Mapping SIP response messages and RFC 3326 reason header and cause
- Locally generated SIP response with RFC 3326 reason header and cause

As specified in RFC 3326, the Oracle Enterprise Session Border Controller sends SIP responses to the softswitch that contain the received Q.850 cause code and the reason.

Though the Oracle Enterprise Session Border Controller can generate RFC 3326 headers, the default behavior for this feature is disabled. Furthermore, the Oracle Enterprise Session Border Controller can receive and pass SIP error messages (4xx, 5xx, and 6xx) that contain the SIP reason header with a Q.850 cause code and reason (as specified in RFC 3326). If the Oracle Enterprise Session Border Controller receives an error message without the Reason header, then the Oracle Enterprise Session Border Controller is not required to insert one.

In calls that require IWF, the Q.850 cause generated in the SIP response are the same as the cause received in the following H.225 messages: Disconnect, Progress, Release, Release Complete, Resume Reject, Status, and Suspend Reject. In addition, the Q.850 cause codes that the Oracle Enterprise Session Border Controller receives in RFC 3326 headers are passed to the H.323 part of the call unmodified; the H.323 call leg uses this cause code for releasing the call.

For interworking calls between SIP and H.323, you can configure:

- Mappings for SIP status codes to Q.850 values
- Mappings for particular Q.850 cause codes to SIP status codes

If it cannot find the appropriate mapping, then the Oracle Enterprise Session Border Controller uses default mappings defined in the Default Mappings table below.

The following describes how the Oracle Enterprise Session Border Controller handles different IWF call scenarios:

- SIP request containing a Reason header—When it receives a request containing a Reason header, the Oracle Enterprise Session Border Controller determines if the request is a SIP BYE or SIP CANCEL message. RFC 3326 states that the Reason header is mainly used for these types of requests. If there is a Reason header and it contains the Q.850 cause value, then the Oracle Enterprise Session Border Controller releases the call on the H.323 side using the specified cause value.
- SIP response—When it receives the error response to an initial SIP INVITE, the Oracle Enterprise Session Border Controller uses its SIP-Q.850 map to determine the Q.850 that it will use to release the call. If there is not a map entry, then the Oracle Enterprise Session Border Controller uses the default mappings shown in the Default Mappings table.
- Active call released from the H.323 side—If an active call is released from the H.323 side, the Oracle Enterprise Session Border Controller checks the outgoing realm (the SIP side) to see if the addition of the Reason header is enabled. If it is, then the Oracle Enterprise Session Border Controller adds the Reason header in the SIP BYE request with the Q.850 value it received from the H.323 side.
- Error during setup of the call on the H.323 side—In the event of an error during setup on the H.323 side of the call, the Oracle Enterprise Session Border Controller needs to send:
  - An error response, if this is a SIP to H.323 call
  - A SIP CANCEL, if this is a H.323 to SIP call and the H.323 side hangs up before the call is answered on the SIP side

    In this case, the Oracle Enterprise Session Border Controller checks to see if adding the Reason header is enabled in the IWF configuration. If it is, then the Oracle Enterprise Session Border Controller adds the Reason header with the Q.850 cause value it received from the H.323 side.
- Call released due to a Oracle Enterprise Session Border Controller error—If the call is released due a Oracle Enterprise Session Border Controller error and adding the Reason header is enabled in the IWF configuration, the error response to the initial INVITE contains the Reason header. The Oracle Enterprise Session Border Controller checks the SIP to Q.850 map configurations to determine whether or not the SIP error response code it is generating is configured. If it is, then the system maps according to the configuration. If if it not, the Oracle Enterprise Session Border Controller derives cause mapping from the default table.

Like the configuration for SIP-only calls that enable this feature, you can set a parameter in the IWF configuration that enables adding the Reason header in the SIP requests or responses.

Default Mappings

This table defines the default mappings the Oracle Enterprise Session Border Controller uses when it cannot locate an appropriate entry that you have configured.

Q.850 Cause Value		SIP Status		Comments
1	Unallocated number	404	Not found	
2	No route to specified transit network	404	Not found	
3	No route destination	404	Not found	
16	Normal calling clearing		BYE message	A call clearing BYE message containing cause value 16 normally results in the sending of a SIP BYE or CANCEL request. However, if a SIP response is to be sent to the INVITE request, the default response code should be used.
17	User busy	486	Busy here	

Q.850 Cause Value		SIP Status		Comments
18	No user responding	408	Request timeout	
19	No answer from the user	480	Temporarily unavailable	
20	Subscriber absent	480	Temporarily unavailable	
21	Call rejected	603	Decline (if location filed in Cause information element indicates user; otherwise 403 Forbidden is used)	
22	Number changed	301	Moved permanently (if information in diagnostic field of Cause information element is suitable for generating SIP Contact header; otherwise 410 Gone is used)	
23	Redirection to new destination	410	Gone	
25	Exchange routing error	483	Too many hops	
27	Destination out of order	502	Bad gateway	
28	Address incomplete	484	Address incomplete	
29	Facility rejected	501	Not implemented	
31	Normal, unspecified	480	Temporarily unavailable	
34	No circuit, channel unavailable	503	Service unavailable	
38	Network out of order	503	Service unavailable	
41	Temporary failure	503	Service unavailable	
42	Switching equipment congestion	503	Service unavailable	
47	Resource unavailable unspecified	503	Service unavailable	
55	Incoming calls barred with CUG	403	Forbidden	
57	Bearer capability not authorized	403	Forbidden	
58	Bearer capability not presently available	503	Service unavailable	
65	Bearer capability not implemented	488	Not acceptable here	
69	Requested facility not implemented	501	Not implemented	

Q.850 Cause Value		SIP Status		Comments
70	Only restricted digital information available	488	Not acceptable here	
79	Service or option not implemented, unspecified	501	Not implemented	
87	User not member of CUG	403	Forbidden	
88	Incompatible destination	503	Service unavailable	
102	Recovery on timer expiry	504	Server time-out	

RFC 3326 Support Configuration

To configure a SIP status to Q.850 Reason with cause mapping:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-q850-map and press Enter.

```
ACMEPACKET(session-router)# sip-q850-map
ACMEPACKET(sip-q850-map)#
```

4. Type entries and press Enter.

```
ACMEPACKET(sip-q850-map)# entries
ACMEPACKET(sip-q850-map-entry)#
```

   From here, you can view the entire menu for the SIP status to Q.850 Reason with cause mapping entries configuration by typing a ?.

5. sip-status—Set the SIP response code that you want to map to a particular Q.850 cause code and reason. There is no default, and the valid range for values is:
   - Minimum—100
   - Maximum—699
6. q850-cause—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
7. q850-reason—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

   To configure a Q.850 cause to a SIP status with reason mapping:

10. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

11. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

12. Type sip-q850-map and press Enter.

```
ACMEPACKET(session-router)# q850-sip-map
ACMEPACKET(q850-sip-map)#
```

13. Type entries and press Enter.

```
ACMEPACKET(q850-sip-map)# entries
ACMEPACKET(q850-sip-map-entry)#
```

From here, you can view the entire menu for the Q.850 cause to a SIP response code with reason mapping entries configuration by typing a ?.

14. q850-cause—Set the Q.850 cause code that you want to map to a SIP status with reason. There is no default.
15. sip-status—Set the SIP response code to which you want to map the Q.850 cause that you set in step 5. There is no default, and the valid range for a value is
    - Minimum—100
    - Maximum—699
16. sip-reason—Set the reason that you want to use with the SIP response code that you specified in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
17. Repeat this process to create the number of local response map entries that you need.

To enable the Oracle Enterprise Session Border Controller to add the Reason header for calls that require IWF:

18. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

19. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

20. Type iwf-config and press Enter.

```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```

21. add-reason-header—Enable this parameter to add the Reason header to IWF calls. The default is disabled. Valid values are:
    - enabled | disabled

IWF Privacy Caller Privacy on Unsecure Networks

This feature enables bi-directional SIP/H.323 IWF support for CPID hiding by using the presentation indicators in the Calling Party Number information element for H.323 signaling, and RFC 3325-based privacy support for SIP signaling. It lets the Oracle Enterprise Session Border Controller insert the P-Asserted-Identity and the Privacy header in the INVITE when the presentation indicator is set to restricted.

The presence, or absence, of P-Asserted-Identity and Privacy headers in the SIP INVITE informs the remote SIP proxy or endpoint to either block or advertise the CPID.

About the Presentation Indicator

When address information represents a telephone number, the relevant information can appear in the Calling Party Number information element (IE). This IE contains the caller's number, information about the number, and presentation and screening indicators found in octet 3a. In order to prevent a calling party number to be passed through, the presentation indicator parameter (octet 3a) in the Calling Party IE must be set to a value other than 00.

In a H.323 to SIP IWF call, octet 3a in the Q.931 message indicates the caller's preference for CPID restriction. If bits 7 and 6 are set to (0 1), the presentation is restricted and the outbound SIP INVITE from the IWF stack must be constructed as such.

H.323 to SIP IWF Call

When the presentation indicator in the calling party IE is set to restricted, the INVITE's From and Contact headers sent from to sipd will be modified according to RFC 3325. When the Oracle Enterprise Session Border Controller

receives calls initiated as H.323, it will recognize the caller's presentation bits as defined in Q.931 and use that information to construct a SIP INVITE in accordance with the user's indicated preference.

- Inclusion of a P-Asserted-Identity header in the INVITE, containing the calling party's CPID and the Oracle Enterprise Session Border Controller's IP address, constructed as a SIP URI (same mechanism used to construct the From-URI today).
- Addition of a Privacy header with its value set to id. This addition indicates to the upstream proxies and gateways that the caller address is to be hidden.

The sipd will either proxy or strip these headers according to RFC 3325, depending on the SIP interface and SIP session agent configurations.

Example 1 SETUP Sent from h323d to Remote H.323 Endpoints

```
Q.931
Protocol discriminator: Q.931
Call reference value length: 2
Call reference flag: Message sent from originating side
Call reference value: 2F62
Message type: SETUP (0x05)
Bearer capability
Information element: Bearer capability
Length: 3
...0 1000 = Information transfer capability: Unrestricted digital information
(0x08)
.00. .... = Coding standard: ITU-T standardized coding (0x00)
1... .... = Extension indicator: last octet
...1 0011 = Information transfer rate: 384 kbit/s (0x13)
.00. .... = Transfer mode: Circuit mode (0x00)
1... .... = Extension indicator: last octet
...0 0101 = User information layer 1 protocol: Recommendation H.221 and H.242
(0x05)
1... .... = Extension indicator: last octet
Display  'jdoe\000'
Information element: Display
Length: 9
Display information: jdoe\000
Calling party number
Information element: Calling party number
Length: 2
.... 0000 = Numbering plan: Unknown (0x00)
.000 .... = Number type: Unknown (0x00)
0... .... = Extension indicator: information continues through the next octet
.... ..00 = Screening indicator: User-provided, not screened (0x00)
.01. .... = Presentation indicator: Presentation restricted (0x01)
1... .... = Extension indicator: last octet
```

Example 2 INVITE from h323d to sipd

The two new headers will be stripped by the sipd when the INVITE is sent to a untrusted SIP proxy or endpoint and will be proxied over to a trusted SIP proxy or end point.

```
INVITE sip:780@192.168.200.6:5060;acme_realm=internal SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5070;branch=z9hG4bKIWF00000510d031s9kou5c0;acme_irealm=external
Contact: "Anonymous"<sip:anonymous@127.0.0.1:5070
GenericID: 7400000@000825010100
Supported: 100rel
From: "Anonymous"<sip:anonymous@anonymous.invalid>;tag=0000004a000d8cc0
To: <sip:780@192.168.200.6:5060
Call-ID: 7f00000113ce0000004a000d88d8@127.0.0.1
CSeq: 2 INVITE
P-Asserted-Identity: "jdoe"<sip:42343@192.168.200.68:5060>
Privacy: id
```

```
Content-Length: 175
Content-Type: application/sdp
v=0
o=IWF 3 3 IN IP4 192.168.1.6
s=H323 Call
c=IN IP4 192.168.1.6
t=0 0
m=audio 5666 RTP/AVP 0 101  18
a=rtpmap:0 PCMU/8000/1
a=rtpmap:101 telephone-event/8000/1
a=fmtp:101 0-15
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
m=video 5668 RTP/AVP 31
a=rtpmap:31 H261/9000/1
```

SIP to H.323

For a SIP to H.323 call, the Oracle Enterprise Session Border Controller must recognize the caller's Privacy request and set the presentation bits accordingly when constructing the outbound RAS/SETUP message. It must check SIP calls for the Privacy header (with value set to id). If this header is present, the SETUP's octet 3a's presentation bits must be set to restricted.

The Oracle Enterprise Session Border Controller does not support any other value for the Privacy header. For those calls, the SETUP will not include a presentation indicator.

Example INVITE from SIP End Point to sipd

```
Apr 21 08:50:38.786 On [0:0]192.168.200.68:5060 received from
192.168.200.6:5062
INVITE sip:800@192.168.200.68:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.6:5062
From: anonymous <sip:anonymous@192.168.200.6:5062>;tag=1
To: sut <sip:800@192.168.200.68:5060
P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062
Privacy: id
Call-ID: 1.1688.192.168.200.6@sipp.call.id
Cseq: 1 INVITE
Contact: sip:anonymous@192.168.200.6:5062
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4       127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
Sample INVITE from sipd to h323d
Apr 21 08:50:38.807 On 127.0.0.1:5070 received from 127.0.0.1:5060
INVITE sip:800@127.0.0.1:5070;acme_sag=sag1;acme_irealm=internal SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK0804o700c0f0t9gpj0g0.1
From: anonymous <sip:anonymous@192.168.200.6:5062>;tag=SDm8kvc01-1
To: sut <sip:800@192.168.200.68:5060
P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062
Privacy: id
Call-ID: SDm8kvc01-083221d8c0fa33f71ae85dd6ed0e4ea4-06ahc21
Cseq: 1 INVITE
Contact: <sip:anonymous@192.168.200.68:5060;transport=udp
Max-Forwards: 69
Subject: Performance Test
```

```
Content-Type: application/sdp
Content-Length: 136
GenericID: 9883100005@000825010100
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4       127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
Sample SETUP sent from h323d to remote H323 EP
Q.931
    Protocol discriminator: Q.931
    Call reference value length: 2
    Call reference flag: Message sent from originating side
    Call reference value: 664D
    Message type: SETUP (0x05)
Bearer capability
        Information element: Bearer capability
        Length: 3
        ...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)
        .00. .... = Coding standard: ITU-T standardized coding (0x00)
        1... .... = Extension indicator: last octet
        ...1 0000 = Information transfer rate: 64 kbit/s (0x10)
        .00. .... = Transfer mode: Circuit mode (0x00)
        1... .... = Extension indicator: last octet
        ...0 0011 = User information layer 1 protocol: Recommendation G.711 A-
law (0x03)
1... .... = Extension indicator: last octet
    Display  'anonymous'
        Information element: Display
        Length: 9
        Display information: anonymous
    Calling party number
        Information element: Calling party number
Length: 2
        .... 0000 = Numbering plan: Unknown (0x00)
        .000 .... = Number type: Unknown (0x00)
        0... .... = Extension indicator: information continues through the
next octet
        .... ..00 = Screening indicator: User-provided, not screened (0x00)
        .01. .... = Presentation indicator: Presentation restricted (0x01)
        1... .... = Extension indicator: last octet
```

IWF Privacy Caller Privacy on Secure Connections

In prior releases, when the H.323 endpoint sends a SETUP with presentation indicator set to allowed, the Oracle Enterprise Session Border Controller does not insert the P-Asserted-Identity in the INVITE. The SIP INVITE needs the P-Asserted-Identity header to support calling line identification presentation (CLIP) to calling line identification restriction (CLIR) in an IP multimedia subsystem (IMS) solution. This feature lets the Oracle Enterprise Session Border Controller insert the P-Asserted-Identity in the INVITE when the presentation indicator is set to allowed.

- CLIP is a service provided to the called party that allows the display of the calling number (caller ID). The user-provided calling number must be transported from the caller to the called party.
- CLIR is a service provided to the calling party that lets it indicate whether or not the calling number is to be displayed to the called party. It sets a calling number presentation indicator to allowed or restricted. Regulations require that network administrations remove the calling number before it is sent to the called party, if the calling party has so requested.

H.323 to SIP IWF

When the Oracle Enterprise Session Border Controller translates incoming H.323 messages to SIP on a secure connection (which means the Oracle Enterprise Session Border Controller can rely on the data sent from the originator); it will translate the information in the H.323 messages into SIP messages as detailed in the following sections.

Calls with Presentation Allowed

When the Oracle Enterprise Session Border ControllerC receives a SETUP from the H.323 domain where presentation is allowed, it generates an INVITE to the SIP domain with the following header. (Presentation is allowed when the calling party's information element presentation indicator (octet 3a) equals 00.)

- P-Asserted-ID: the userpart should be derived from the Calling Party Number Information Element digits.

H.323 to SIP

When h323d receives a SETUP with the calling party's information element presentation indicator set to allowed, the Oracle Enterprise Session Border Controller will add the P-Asserted-Identity header to the INVITE. The P-Asserted-Identity is very similar to the FROM header, except for the tag.

Sample SETUP sent from h323d to Remote H323 Endpoints

```
Q.931
Protocol discriminator: Q.931
Call reference value length: 2
Call reference flag: Message sent from originating side
Call reference value: 2F62
Message type: SETUP (0x05)
Bearer capability
Information element: Bearer capability
Length: 3
...0 1000 = Information transfer capability: Unrestricted digital information
(0x08)
.00. .... = Coding standard: ITU-T standardized coding (0x00)
1... .... = Extension indicator: last octet
...1 0011 = Information transfer rate: 384 kbit/s (0x13)
.00. .... = Transfer mode: Circuit mode (0x00)
1... .... = Extension indicator: last octet
...0 0101 = User information layer 1 protocol: Recommendation H.221 and H.242
(0x05)
1... .... = Extension indicator: last octet
Display  'jdoe\000'
Information element: Display
Length: 9
Display information: jdoe\000
Calling party number: '42343'
Information element: Calling party number
Length: 6
.... 1001 = Numbering plan: Private numbering (0x09)
.110 .... = Number type: Abbreviated number (0x06)
0... .... = Extension indicator: information continues through the next octet
.... ..00 = Screening indicator: User-provided, not screened (0x00)
.00. .... = Presentation indicator: Presentation allowed (0x00)
1... .... = Extension indicator: last octet
Calling party number digits: 42343
```

SIP to H.323

When the sipd receives an INVITE with the P-Asserted-Identity header but without the Privacy header, the Oracle Enterprise Session Border Controller will set the presentation indicator to allowed in H.323's SETUP.

When the Privacy header is present with the value id, the presentation indicator will be set to restricted. The Oracle Enterprise Session Border Controller does not support any other value for the Privacy header and so for those call flows, the presentation indicator will be absent in the SETUP.

Example 1 INVITE from sip EP to sipd

```
Apr 20 04:43:54.220 On [0:0]192.168.200.68:5060 received from
192.168.200.6:5062
INVITE sip:800@192.168.200.68:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.6:5062
From: sipp <sip:7789@192.168.200.6:5062>;tag=1
To: sut <sip:800@192.168.200.68:5060>
P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062>
Call-ID: 1.1336.192.168.200.6@sipp.call.id
Cseq: 1 INVITE
Contact: sip:7789@192.168.200.6:5062
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
^M
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4       127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Example INVITE from sipd to h323d

```
Apr 20 04:43:54.240 On 127.0.0.1:5070 received from 127.0.0.1:5060
INVITE sip:800@127.0.0.1:5070;acme_sag=sag1;acme_irealm=internal SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK000c0210385hv9gpt001.1
From: sipp <sip:7789@192.168.200.6:5062>;tag=SDk0jpc01-1
To: sut <sip:800@192.168.200.68:5060>
Call-ID: SDk0jpc01-8e15e11e7f9a20523462972843c7e579-06ahc21
Cseq: 1 INVITE
Contact: <sip:7789@192.168.200.68:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
GenericID: 160400004@000825010100
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4       127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
Sample SETUP sent from h323d to remote H323 EP
Q.931
    Protocol discriminator: Q.931
Call reference value length: 2
    Call reference flag: Message sent from originating side
    Call reference value: 664D
    Message type: SETUP (0x05)
    Bearer capability
        Information element: Bearer capability
        Length: 3
        ...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)
        .00. .... = Coding standard: ITU-T standardized coding (0x00)
        1... .... = Extension indicator: last octet
```

```
        ...1 0000 = Information transfer rate: 64 kbit/s (0x10)
        .00. .... = Transfer mode: Circuit mode (0x00)
        1... .... = Extension indicator: last octet
        ...0 0011 = User information layer 1 protocol: Recommendation G.711 A-
law (0x03)
        1... .... = Extension indicator: last octet
    Display  'sipp'
        Information element: Display
        Length: 4
        Display information: sipp
    Calling party number: '7789'
        Information element: Calling party number
        Length: 6
        .... 1001 = Numbering plan: Private numbering (0x09)
        .110 .... = Number type: Abbreviated number (0x06)
        0... .... = Extension indicator: information continues through the
next octet
.... ..00 = Screening indicator: User-provided, not screened (0x00)
        .00. .... = Presentation indicator: Presentation all 1... .... =
Extension indicator: last octet
        Calling party number digits: 7789
```

IWF Privacy Extensions for Asserted Identity in Untrusted Networks

For IWF privacy, the Oracle Enterprise Session Border Controller supports:

- IWF caller privacy on unsecure networks—A variant of RFC 3325, where the P-Asserted-Id is inserted when the presentation indicator is set to allowed. This feature enables bi-directional SIP/H.323 IWF support for CPID hiding by using the presentation indicators in the Calling Party Number information element forH.323 signaling, and RFC 3325-based privacy support for SIP signaling. It allows the Oracle Enterprise Session Border Controller to insert the P-Asserted-Identity and the Privacy header in the INVITE when the presentation indicator is set to restricted.

  The presence, or absence, of P-Asserted-Identity and Privacy headers in the SIP INVITE informs the remote SIP proxy or endpoint to either block or advertise the CPID.
- IWF caller privacy on secure connections—When the H.323 endpoint sends a SETUP with presentation indicator set to allowed, the Oracle Enterprise Session Border Controller does not insert the P-Asserted-Identity in the INVITE. The SIP INVITE needs the P-Asserted-Identity header to support calling line identification presentation (CLIP) to calling line identification restriction (CLIR) in an IP multimedia subsystem (IMS) solution. This feature allows the Oracle Enterprise Session Border Controller to insert the P-Asserted-Identity in the INVITE when the presentation indicator is set to allowed.

Now the Oracle Enterprise Session Border Controller supports an enhancement to IWF caller privacy where the P-Preferred-Identity is inserted instead of the P-Asserted-Identity.

In this implementation, when the incoming H.323 Setup message has a presentation indicator set to restricted and the ingress H.323 session agent has the new PPreferredId option configured, the Oracle Enterprise Session Border Controller sends the Privacy header with P-Preferred-Identity (instead of P-Asserted-Identity).

IWF Call Originating in H.323

This section shows an example H.323 Setup that arrives from an H.323 endpoint, and how the Oracle Enterprise Session Border Controller adds the P-Preferred-Identity header (which has calling party number information) and the Privacy header to the SIP INVITE.

Sample H.323 Setup from a Remote Endpoint

```
Q.931
    Protocol discriminator: Q.931
```

```
    Call reference value length: 2
    Call reference flag: Message sent from originating side
    Call reference value: 2FB6
    Message type: SETUP (0x05)
    Bearer capability
        Information element: Bearer capability
        Length: 3
        ...0 1000 = Information transfer capability: Unrestricted digital
information (0x08)
        .00. .... = Coding standard: ITU-T standardized coding (0x00)
        1... .... = Extension indicator: last octet
        ...1 0011 = Information transfer rate: 384 kbit/s (0x13)
        .00. .... = Transfer mode: Circuit mode (0x00)
        1... .... = Extension indicator: last octet
...0 0101 = User information layer 1 protocol: Recommendation H.221 and H.242
(0x05)
        1... .... = Extension indicator: last octet
    Display  'rdoe\000'
        Information element: Display
        Length: 9
        Display information: jdoe\000
    Calling party number: '42343'
        Information element: Calling party number
        Length: 6
        .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .000 .... = Number type: Unknown (0x00)
        0... .... = Extension indicator: information continues through the
next octet
        .... ..00 = Screening indicator: User-provided, not screened (0x00)
        .01. .... = Presentation indicator: Presentation restricted (0x01)
        1... .... = Extension indicator: last octet
        Calling party number digits: 42343
E.164 Calling party number digits: 42343
    Called party number: '780'
        Information element: Called party number
        Length: 4
        .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .000 .... = Number type: Unknown (0x00)
        1... .... = Extension indicator: last octet
        Called party number digits: 780
        E.164 Called party number digits: 780
User-user
        Information element: User-user
        Length: 161
        Protocol discriminator: X.208 and X.209 coded user information
```

Sample SIP INVITE from the SBC to a SIP Endpoint

```
Aug 29 15:46:25.214 On [0:0]192.168.200.68:5060 sent to 192.168.200.6:5060
INVITE sip:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.68:5060;branch=z9hG4bK6810pr20205h2akqe381.1
Contact: "Anonymous"<sip:anonymous@192.168.200.68:5060;transport=udp>
Supported: 100rel
From:
"Anonymous"<sip:anonymous@anonymous.invalid>;tag=SDfd9sa01-000000ba00023280
To: <sip:780@192.168.200.6:5060>
Call-ID: SDfd9sa01-6f93292521b83a0980647f34451c5afd-06ahc21
CSeq: 2 INVITE
P-Preferred-Identity: "rdoe"<sip:42343@192.168.200.68:5060>
<b>Privacy: id<\b>
Content-Length: 180
Content-Type: application/sdp
Max-Forwards: 70
v=0
```

```
o=IWF 5 5 IN IP4 192.168.200.5
s=H323 Call
c=IN IP4 192.168.200.65
t=0 0
m=audio 5010 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1
m=video 5014 RTP/AVP 31
a=rtpmap:31 H261/9000/1
```

Before You Configure

Before you configure your Oracle Enterprise Session Border Controller to support this feature, note the following considerations:

- The ingress H.323 session agent cannot be configured with the NoPAssertedId option
- For use in Release 4.1.1 and higher, the global SIP configuration should be configured with the disable-ppi-to-pai option; the older disable-privacy option will also work

P-Preferred-Identity Configuration

To enable the inclusion of P-Preferred-Identity:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. Select the session agent where you want to apply this feature.

   ```
   ACMEPACKET(session-agent)# select
   <hostname>:
   1: 204.12.60.5      realm=private
   2: 124.21.5.3       realm=public
   selection:1
   ACMEPACKET(session-agent)#
   ```

5. options—Set the options parameter by typing options, a Space, the option name preceded by a plus sign (+) (PPreferredId), and then press Enter.

   ```
   ACMEPACKET(realm-config)# options +PPreferredId
   ```

   If you type options PPreferredId, you will overwrite any previously configured options. In order to append the new option to the session agent's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

IWF Privacy for Business Trunking

The Oracle Enterprise Session Border Controller supports IWF Privacy: Caller Privacy on Unsecure Networks and IWF Privacy: Caller Privacy on Secure Connections, but IWF Privacy for Business Trunking, supports the case where SIP and H.323 PBXs are connected to the core IMS system. Traffic originated at the IP PBXs terminates either at other PBXs or at the PSTN, and includes the possibility of accepting incoming traffic from the PSTN. CLIP and CLIR must be supported for calls in either direction for calls that require interworking between SIP and H.323. Unlike the two features described above, this new feature supports the fact that only a network-based application server has sufficient privilege to assert the identity of the calling party.

Thus, for this feature, the Oracle Enterprise Session Border Controller does not force privacy. Instead, the implemented feature assumes that the H.323 session agent is an IP PB X, and the Oracle Enterprise Session Border Controller only indicates to the SIP core that privacy is being requested. In other words, the Oracle Enterprise Session Border Controller is not required to interwork the H.323 presentation indicator parameter to RFC 3325 by including the P-Asserted-Identity header. The indication to the SIP core that privacy is being requested excludes identity assertion.

You configure this feature using two session agent options:

- allowCPN—Set in the egress H.323 session agent, allows the Oracle Enterprise Session Border Controller to send the calling party number information element (IE), even when the presentation indicator is set to restricted.
- NoPAssertedId—Set in the ingress H.323 session agent; when the incoming SETUP message has the presentation indicator is set to restricted, instructs the Oracle Enterprise Session Border Controller to send a Privacy header without the P-Asserted-Identity and not to make the From header anonymous.

A Call Originating in H.323

This section describes for the IWF Privacy for Business trunking feature works for a call originating in H.323 that requires interworking to SIP.

When the Oracle Enterprise Session Border Controller receives an H.323 SETUP with a presentation indicator of the calling party information element (IE) is set to restricted and this SETUP was received from a session agent is configured with the NoPAssertedID option, the Oracle Enterprise Session Border Controller only adds the Privacy header with the value ID. In this case, there will be no P-Asserted-Identity and the From header will contain the calling Party information that was extracted from the callingPartyIE. The Oracle Enterprise Session Border Controller assumes that the PBX will send the callingPartyNumber in the IE, even though it would like to have the calling party number restricted.

Sample SETUP Message from an H.323 Endpoint

```
Q.931
    Protocol discriminator: Q.931
    Call reference value length: 2
    Call reference flag: Message sent from originating side
    Call reference value: 2FB6
    Message type: SETUP (0x05)
    Bearer capability
        Information element: Bearer capability
Length: 3
        ...0 1000 = Information transfer capability: Unrestricted digital
information (0x08)
        .00. .... = Coding standard: ITU-T standardized coding (0x00)
        1... .... = Extension indicator: last octet
        ...1 0011 = Information transfer rate: 384 kbit/s (0x13)
        .00. .... = Transfer mode: Circuit mode (0x00)
        1... .... = Extension indicator: last octet
        ...0 0101 = User information layer 1 protocol: Recommendation H.221
and H.242 (0x05)
        1... .... = Extension indicator: last octet
    Display  'jdoe\000'
        Information element: Display
Length: 9
        Display information: jdoe\000
    Calling party number: '42343'
        Information element: Calling party number
        Length: 6
        .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .000 .... = Number type: Unknown (0x00)
        0... .... = Extension indicator: information continues through the
next octet
        .... ..00 = Screening indicator: User-provided, not screened (0x00)
        .01. .... = Presentation indicator: Presentation restricted (0x01)
```

```
        1... .... = Extension indicator: last octet
        Calling party number digits: 42343
        E.164 Calling party number digits: 42343
    Called party number: '780'
        Information element: Called party number
        Length: 4
        .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .000 .... = Number type: Unknown (0x00)
        1... .... = Extension indicator: last octet
        Called party number digits: 780
        E.164 Called party number digits: 780
User-user
        Information element: User-user
        Length: 161
        Protocol discriminator: X.208 and X.209 coded user information
```

Sample INVITE from the Oracle Enterprise Session Border Controller to the SIP Endpoint

```
May  5 15:11:51.996 On [0:0]192.168.200.68:5060 sent to 192.168.200.6:5060
INVITE sip:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.68:5060;branch=z9hG4bK00020a20eg11s94pg700.1
Contact: "jdoe"<sip:42343@192.168.200.68:5060;transport=udp>
Supported: 100rel
From: "jdoe"<sip:42343@192.168.200.68:5060>;tag=SDetur801-00000194000e2ce8
To: <sip:780@192.168.200.6:5060>
Call-ID: SDetur801-231c7b30909ca525ce12cbfeb57754ea-06ahc21
CSeq: 2 INVITE
Privacy: id
Content-Length: 231
Content-Type: application/sdp
Max-Forwards: 70
v=0
o=IWF 2 2 IN IP4 192.168.200.65
s=H323 Call
c=IN IP4 192.168.200.65
t=0 0
m=audio 5004 RTP/AVP 8 0
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000/1
m=video 5006 RTP/AVP 31 34
a=rtpmap:31 H261/8000
a=rtpmap:34 H263/9000/1
```

A Call Originating in SIP

This section describes for the IWF Privacy for Business trunking feature works for a call originating in SIP that requires interworking to H.323.

When the Oracle Enterprise Session Border Controller receives a SIP INVITE with a Privacy header that has the value ID, it sets the presentation indicator to restricted in the corresponding H.323 SETUP message. If the H.323 session agent is configured with the allowCPN option, the Oracle Enterprise Session Border Controllersends the display IE and the calling party number to the H.323 session agent. If that option is not set in the H.323 session agent, then the Oracle Enterprise Session Border Controller reverts to its default behavior, which is to not to send the display IE and to hide the calling party number.

Sample INVITE from a SIP Endpoint to the Oracle Enterprise Session Border Controller

```
May  5 14:41:54.513 On [0:0]192.168.200.68:5060 received from
192.168.200.6:5060
INVITE sip:800@192.168.200.68:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.6:5060
```

```
From: sipp <sip:sipp@192.168.200.6:5060>;tag=1
To: sut <sip:800@192.168.200.68:5060>
Call-ID: 1.3068.192.168.200.6@sipp.call.id
Cseq: 1 INVITE
Contact: sip:sipp@192.168.200.6:5060
Privacy: id
P-Asserted-Identity: sipp <sip:1234@192.168.200.6:5060>
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4       127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Sample SETUP from the Oracle Enterprise Session Border Controller to the H.323 Endpoint

```
Q.931
    Protocol discriminator: Q.931
    Call reference value length: 2
    Call reference flag: Message sent from originating side
    Call reference value: 44B0
    Message type: SETUP (0x05)
    Bearer capability
        Information element: Bearer capability
        Length: 3
        ...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)
        .00. .... = Coding standard: ITU-T standardized coding (0x00)
        1... .... = Extension indicator: last octet
        ...1 0000 = Information transfer rate: 64 kbit/s (0x10)
        .00. .... = Transfer mode: Circuit mode (0x00)
        1... .... = Extension indicator: last octet
        ...0 0011 = User information layer 1 protocol: Recommendation G.711 A-
law (0x03)
        1... .... = Extension indicator: last octet
    Display  'sipp'
        Information element: Display
        Length: 4
        Display information: sipp
Calling party number: '1234'
        Information element: Calling party number
Length: 6
        .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .010 .... = Number type: National number (0x02)
        0... .... = Extension indicator: information continues through the
next octet
        .... ..00 = Screening indicator: User-provided, not screened (0x00)
        .01. .... = Presentation indicator: Presentation restricted (0x01)
        1... .... = Extension indicator: last octet
        Calling party number digits: 1234
        E.164 Calling party number digits: 1234
    Called party number: '800'
        Information element: Called party number
        Length: 4
.... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
        .010 .... = Number type: National number (0x02)
        1... .... = Extension indicator: last octet
        Called party number digits: 800
        E.164 Called party number digits: 800
```

```
    User-user
        Information element: User-user
        Length: 159
        Protocol discriminator: X.208 and X.209 coded user information
```

allowCPN Configuration

You can set both of these options in the same H.323 session agent.

To set the allowCPN option for an H.323 session agent:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ```

4. Use the ACLI select command so that you can work with the session agent configuration to which you want to add this option.

   ```
   ACMEPACKET(session-agent)# select
   ```

5. options—Set the options parameter by typing options, a Space, the option name allowCPN with a plus sign in front of it, and then press Enter.

   ```
   ACMEPACKET(session-agent)# options +allowCPN
   ```

   If you type options allowCPN (without the plus sign), you will overwrite any previously configured options. In order to append the new option to the session-agent's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

   To set the NoPAssertedId option for an H.323 session agent:

7. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

8. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

9. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ```

10. Use the ACLI select command so that you can work with the session agent configuration to which you want to add this option.

    ```
    ACMEPACKET(session-agent)# select
    ```

11. options—Set the options parameter by typing options, a Space, the option name NoPAssertedId with a plus sign in front of it, and then press Enter.

    ```
    ACMEPACKET(session-agent)# options +NoPAssertedId
    ```

    If you type options NoPAssertedId (without the plus sign), you will overwrite any previously configured options. In order to append the new option to the session-agent's options list, you must prepend the new option with a plus sign as shown in the previous example.

12. Save and activate your configuration.

Trunk Group URIs

The Oracle Enterprise Session Border Controller's trunk group URI feature, applicable for SIP and IWF signaling services, enables the capabilities related to trunk groups that are described in this section. This implementation follows the IPTEL draft Representing Trunk Groups in Tel/SIP Uniform Resource Identifiers (URIs) (draft-ietf-iptel-trunk-group-06.txt), and also supports more customized approaches.

- For a typical access call flow scenario, when the calling party's call arrives at the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller formulates a SIP INVITE message that it sends to a softswitch. The Oracle Enterprise Session Border Controller now supports a new URI contact parameter in the SIP request message so that service providers need to be able to:
  - Determine from where the Oracle Enterprise Session Border Controller received the call
  - Signal information about the originating gateway from a Oracle Enterprise Session Border Controller to a softswitch (e.g., an incoming trunk group or a SIP gateway to a Oracle Enterprise Session Border Controller)
- This feature supports the signaling of routing information to the Oracle Enterprise Session Border Controller from network routing elements like softswitches. This information tells the Oracle Enterprise Session Border Controller what egress route (or outgoing trunk groups) it should choose for terminating next hops/gateways. For this purpose, new SIP URI parameters in the Request-URI are defined. Additional URI parameters include the network context to identify the network in which the originating or terminating gateway resides.
- Especially important for large business applications, this feature can free Oracle Enterprise Session Border Controller resources by reducing the number of local policy, session agent, and session agent group configurations. By enabling the trunk group URI feature, the Oracle Enterprise Session Border Controller instead uses a routing scheme based on signaled SIP URI information.

Terminology

The following IPTEL terms are used in the descriptions of and instructions for how to configure this feature:

- Trunk—In a network, a communication path connecting two switching systems used in the establishment of an end-to-end connection; in selected applications, it may have both its terminations in the same switching system
- Trunk group—A set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable except where sub-grouped
- Trunk group name—Provides a unique identifier of the trunk group; referred to as tgrp
- Trunk group context—Imposes a namespace by specifying a domain where the trunk groups are; also referred to simply as context

Trunk Group URI Parameters

Trunk group URI parameters identify originating and terminating trunk group information in SIP requests.

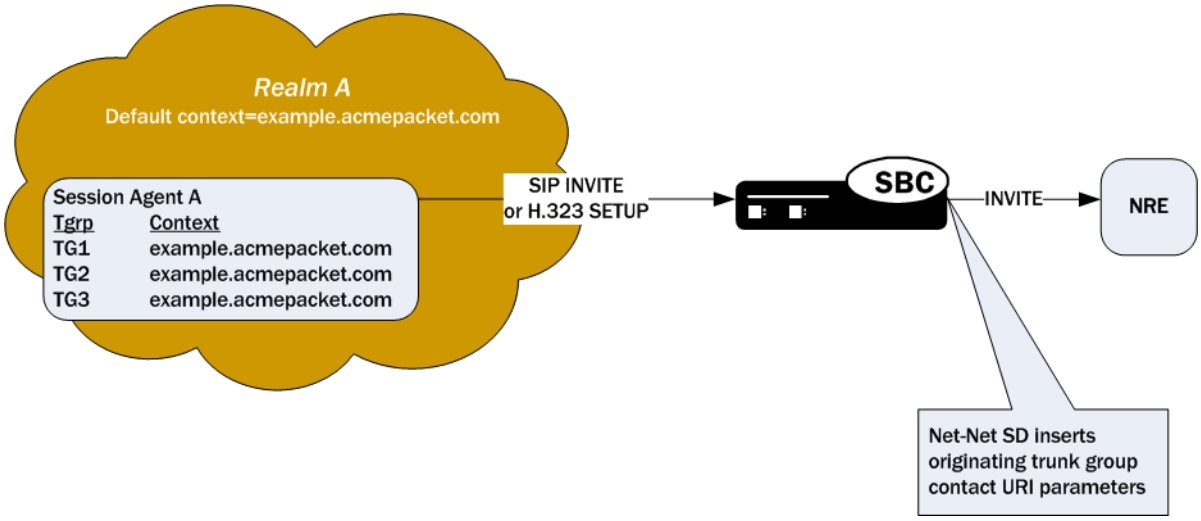
In the absence of official SIP standards for transporting trunk groups between signaling elements, the Oracle Enterprise Session Border Controllerallows you to define URI parameters for use with originating and terminating trunk group URIs.

Originating Trunk Group URI Parameters and Formats

You can configure session agents and session agents groups on the Oracle Enterprise Session Border Controller to insert trunk group URI parameters in the SIP contact header. When SIP gateways comply with the IPTEL draft, they include the originating URI parameter in the SIP contact header. For those SIP and H.323 gateways that are not compliant, the Oracle Enterprise Session Border Controller inserts SIP trunk group URI parameters on the gateway's behalf.

When there are no applicable session agent or session agent group configurations, the Oracle Enterprise Session Border Controller uses the source IP address of the endpoint or gateway as the trunk group name (tgrp) parameter in the originating trunk group URI.

The following diagram shows a scenario where the Oracle Enterprise Session Border Controller inserts originating trunk group URI parameters.



There are two available formats for the originating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (identifier of the specific trunk group) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

   - tgrp=”trunk group name”
   - trunk-context=”network domain”

   The URI BNF for would appear as it does in the example directly below, where the tgrp is tg55 and the trunk-context is trunk-context is telco.example.com:

   ```
   tel:+15555551212;tgrp=tg55;trunk-context=telco.example.com
   ```

2. The second format is customized specifically for access URIs and contains two provisioned parameters: tgrp (or tgname) and context (or provstring). This appears as tgrp.context, where these definitions apply:

   - tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
   - context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

   This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

   ```
   custsite2NY-00020.type2.voip.carrier.net
   ```

   where the tgrp is custsite2NY-00020, and the context is type2.voip.carrier.net.

   The BNF for an access URI conforms to the following:

   SIP-URI = "sip:" [userinfo] hostport uri-parameters [headers]

   uri-parameters = *(";" uri-parameter)

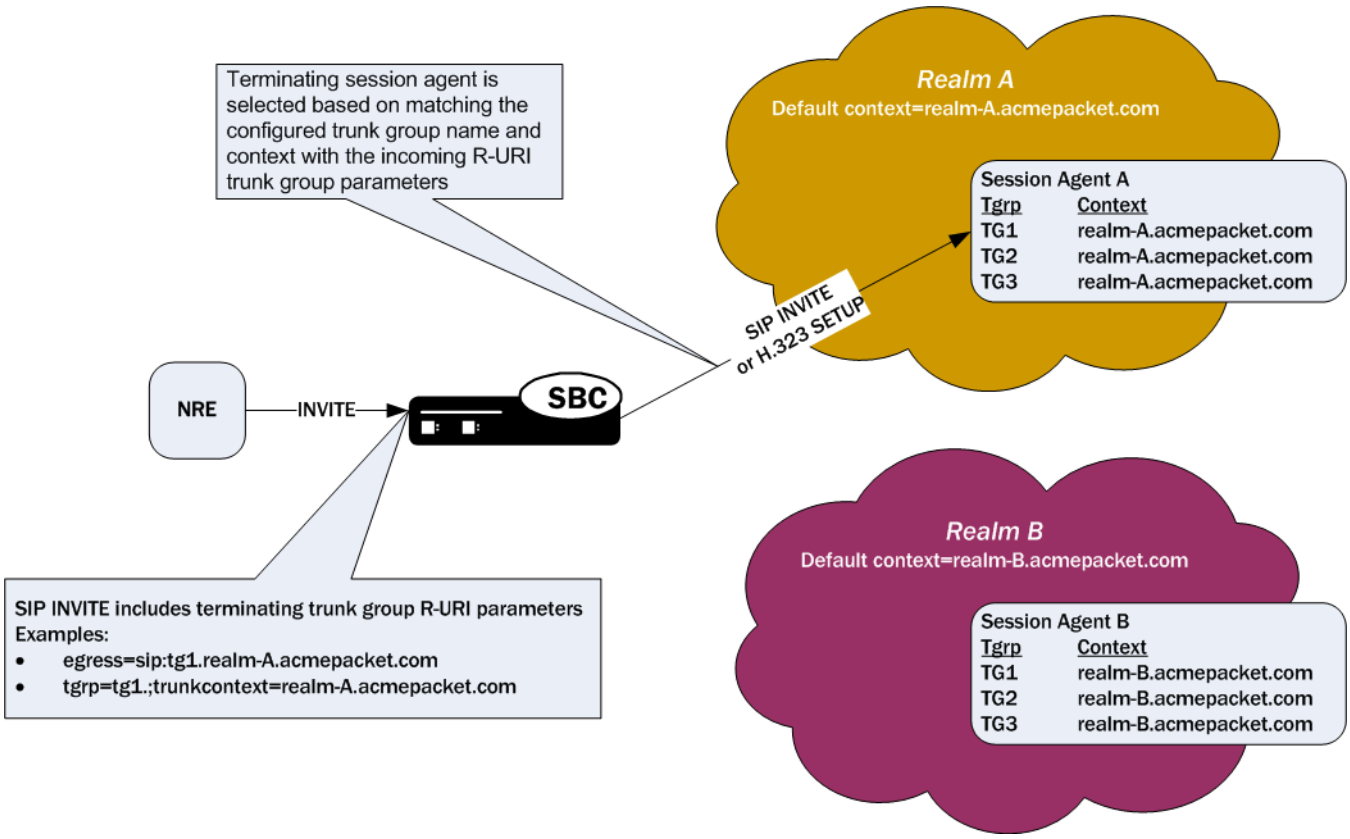
   uri-parameter = transport-param / user-param / method-param

   / ttl-param / maddr-param / lr-param / other-param

   ```
   other-param = accessid / pname [ '=' pvalue ]
   accessid = "access=" accessURI
   accessURI = scheme tgname ["." provstring]
   scheme = "sip:" / token
   tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
       alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
   provstring = *(domain ".") toplabel # up to 24 characters
   toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
   domain = alphanum/ alphanum *( alphanum / "-" ) alphanum
   ```

Terminating Trunk Group URI Parameters and Formats

Terminating trunk group URI parameters appear in the R-URI, and they can be included in by a network routing element to instruct the Oracle Enterprise Session Border Controller which egress trunk groups to use. By matching the trunk group URI parameter with configured session agents or session agent groups, the Oracle Enterprise Session Border Controller can locate the terminating gateway. The trunk group name can also be expressed as the IP address of the terminating gateway.



In the absence of official SIP standards for transporting trunk groups between signaling elements, the Oracle Enterprise Session Border Controller allows you to define the URI parameters used in terminating trunk groups.

There are two available formats for the terminating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (which can be either a trunk group name or an IP address) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

   - tgrp=”trunk group name”
   - trunk-context=”network domain”

   An example R-URI with terminating trunk group parameters appears as follows, where the tgrp is TG2-1 and the context is isp.example.net@egwy.isp.example.net:

   ```
   INVITE sip:+15555551212;tgrp=TG2-1;trunk-
   context=isp.example.net@egwy.isp.example.net SIP/2.0
   ```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: tgrp (or tgname) and context (or tgdomain). This appears as tgrp.context (or tgname.tgdomain), where definitions apply:

   - tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
   - context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```
SIP-URI = "sip:" [userinfo ] hostport uri-parameters [headers ]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
            / ttl-param / maddr-param / lr-param / other-param
other-param = egressid  / pname [ '=' pvalue ]
egressid = "egress=" egressURI
egressURI = scheme tgname ["." tgdomain]
scheme = "sip:" / token
tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
    alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
tgdomain = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
domain = alphanum/ alphanum *( alphanum / "-" ) alphanum
```

For all trunk group URI support, you must set the appropriate parameters in the SIP manipulations configuration and in the session agent or session agent group configurations.

In the originating trunk group URI scenario, a call arrives at the Oracle Enterprise Session Border Controller from a configured session agent or session agent group. If this session agent or session agent group has the appropriate trunk group URI parameters and inbound manipulation rules configured, the Oracle Enterprise Session Border Controller then looks to the SIP manipulations configuration and add the trunk group URI information according to those rules. Those rules tell the Oracle Enterprise Session Border Controller where and how to insert the trunk group URI information, and the Oracle Enterprise Session Border Controller forwards the call.

In the terminating trunk group scenario, a call arrives at the Oracle Enterprise Session Border Controller from, for instance, a call agent. This call contains information about what trunk group to use. If the information matches a session agent or session agent group that has outbound manipulation rules configured, the Oracle Enterprise Session Border Controller will then look up the SIP manipulations configuration and strip information according to those rules. Those rules tell the Oracle Enterprise Session Border Controller where and how to remove the information, and the Oracle Enterprise Session Border Controller forwards the call.

SIP Header and Parameter Manipulation

SIP header and parameter manipulation is its own configuration where you can set up rules for the addition, removal, and modification of a SIP header or the elements of a SIP header. For example, you can set up the configuration to add a URI parameter to the URI in a SIP header or replace an FQDN with in IP address. For trunk group URI support, this configuration tells the Oracle Enterprise Session Border Controller where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

These manipulations can be applied at the realm or at the session agent level.

Trunk Group Routing

You can configure SIP interfaces (using the ACLI term-tgrp-mode parameter) to perform routing based on the trunk group information received in SIP requests. There are three options: none, IPTEL, and egress URI.

- If you leave this parameter set to none (its default), the Oracle Enterprise Session Border Controller will not look for or route based on terminating trunk group URI parameters
- When you set this parameter to either iptel or egress-uri and the incoming request has the trunk group parameter of this type (IPTEL or egress URI), the Oracle Enterprise Session Border Controller will select the egress next hop by matching the “tgrp” and trunk context with a configured session agent or session agent group.

  If the received terminating trunk group URI parameters include an IP address, the egress next hop is the IP address specified. The Oracle Enterprise Session Border Controller determines the egress realm by matching the trunk context it receives with the trunk context you configure for the realm.
- If the incoming request does not have trunk group parameters or it does not have trunk group parameters of the type that you configure, the Oracle Enterprise Session Border Controller uses provisioned procedures and/or local policy for egress call routing.

The Oracle Enterprise Session Border Controller returns errors in these cases:

- If the terminating trunk group URI parameters do not identify a local Oracle Enterprise Session Border Controllersession agent or session agent group, then the Oracle Enterprise Session Border Controller returns a SIP final response of 488 Not Acceptable Here.
- If the Oracle Enterprise Session Border Controller receives a SIP INVITE with terminating trunk group URI parameters that do not match the specified syntax, the Oracle Enterprise Session Border Controllerreturns a 400 final response with the reason phrase Bad Egress=Parameters.

Trunk Group URIs and SIP Registration Caching

For calls where SIP registration caching is used, you will need to set certain parameters that enable the Oracle Enterprise Session Border Controller to preserve trunk group URI parameters on the outgoing side.

- For SIP-H.323 calls requiring IWF, you set the preserve-user-info-sa option in the session agent configuration.

Trunk Group URI Configuration

Before you configure your Oracle Enterprise Session Border Controller to support trunk group URIs, you need to determine:

- How you want to manipulate SIP headers (entered in the SIP header manipulations configuration)
- For terminating trunk group routing, the trunk group mode you want to use (none, IPTEL, or egress URI); this decides routing based on trunk group information
- The trunk group name and context to use entered in a session agent or session agent group configuration
- Whether you are using originating or terminating trunk group URIs (entered in the session agent configuration)
- The trunk group context for use in a realm configuration, in case the trunk group name in the session agent or session agent group does not have a context

Configuring SIP Manipulations

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The new-value parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are $TRUNK_GROUP and $TRUNK_GROUP_CONTEXT

Setting the Trunk Group URI Mode for Routing

To set the mode for routing for terminating trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. term-tgrp-mode—Set the mode that you want to use for routing for terminating trunk group URIs. The default value is none. Valid values are:

   - none—Disables routing based on trunk groups
   - iptel—Uses trunk group URI routing based on the IPTEL formats
   - egress-uri—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. out-manipulationid—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the Oracle Enterprise Session Border Controller via this session agent. There is no default.
5. in-manipulationid—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic entering the Oracle Enterprise Session Border Controller via this session agent. There is no default.
6. trunk-group—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Oracle Enterprise Session Border Controller will use the one you set in the realm for this session agent.

   Your ACLI entries for this list must one of these formats: tgrp:context or tgrp.context.

   To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

   ```
   ACACMEPACKETMEPACKET(session-agent)# trunk-group (tgrp1:context1
   tgrp2:context2)
   ```

7. options—If you want to configure trunk group URIs for SIP-H.323 calls that use the IWF and you are using SIP registration caching, you might need to add the preserve-user-info-sa to your list of session agent options.

   If you are adding this option to a new session agent, you can just type options, a Space, and preserve-user-info-sa.

   If are adding this to an existing session agent, you must type a plus (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +preserve-user-info-sa.

Configuring a Session Agent Group for Trunk Group URIs

In a session agent group, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent group for trunk group URIs:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-group and press Enter.

   ```
   ACMEPACKET(session-router)# session-group
   ACMEPACKET(session-agent-group)#
   ```

4. trunk-group—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Oracle Enterprise Session Border Controller will use the one you set in the realm for this session agent group.

Your ACLI entries for this list must take one of these formats: tgrp:context or tgrp.context.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent-group)# trunk-group (tgrp1:context1 tgrp2:context2)
```

Setting a Trunk Group Context in a Realm

You can set trunk group contexts at the realm level, which will be used by all session agents and session agent groups if there is no context specified in their configurations.

The realm trunk group URI context accommodates the IPTEL and the custom format.

To configure a trunk group context for a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. trunk-context—Enter the trunk group context to use for this realm. There is no default.

Using this Feature with SIP Interface Registration Caching

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the preserve-user-info option for that SIP interface.

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-group and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. options—Add support for trunk group URIs with SIP interface that uses registration caching.

   If you are adding this option to a new SIP interface, you can just type options, a Space, and preserve-user-info.

   If are adding this to an existing SIP interface, you must type a plus (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a plus sign: options +preserve-user-info.

Example 1 Adding Originating Trunk Group Parameters in IPTEL Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTEL format.

```
sip-manipulation
        name                            add_iptel
        header-rule
                name                            contact
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            tgrp
                        type                            uri-user-param
```

```
                        action                          add
                        match-val-type                  any
                        match-value
                        new-value                       $TRUNK_GROUP
element-rule
                        name                            trunk-context
                        type                            uri-user-param
                        action                          add
                        match-val-type                  any
                        match-value
                        new-value                       $TRUNK_GROUP_CONTEXT
```

Example 1 Adding Originating Trunk Group Parameters in Custom Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in custom format.

```
sip-manipulation
        name                            add_att
        header-rule
                name                            contact
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            egressURI
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        match-value
                        new-value                       "sip:"+$TRUNK_GROUP
+"."+$TRUNK_GROUP_CONTEXT
```

Example 2 Removing IPTEL Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove IPTEL trunk groups names.

```
sip-manipulation
        name                            strip_iptel
        header-rule
                name                            request-uri
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            tgrp
                        type                            uri-user-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
element-rule
                        name                            trunk-context
                        type                            uri-user-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
```

Example 3 Removing Custom Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove custom trunk groups names.

```
sip-manipulation
        name                            strip_egress
        header-rule
                name                            request-uri
                action                          manipulate
                match-value
                msg-type                        any
                element-rule
                        name                            egressURI
                        type                            uri-param
                        action                          delete-element
                        match-val-type                  any
                        match-value
                        new-value
```

Configuring SIP Manipulations

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The new-value parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are $TRUNK_GROUP and $TRUNK_GROUP_CONTEXT

IWF COLP COLR Support

When you enable the connected line identity presentation (COLP) and connected line identity restriction (COLR) feature for calls being translated between SIP and H.323, the Oracle Enterprise Session Border Controller converts the H.323 Connected Number Information element (IE) to the SIP P-Asserted-Identity (PAI) header and vice versa.

When there is no Q.931 Connected Number IE, the Oracle Enterprise Session Border Controller converts the H.225 Connected Address alias (either E.164 or Public Party Number).

This section describes show the IWF COLP/COLR feature works for IWF calls that originate in SIP and are translated to H.323, and for calls that originate in H.323 and are translated to SIP.

SIP to H.323 Calls

For this type of call, the Oracle Enterprise Session Border Controller checks the Connect that it receives for a Q.931 Connected Number IE. If it does not find one, then it continues by checking for H.225 Connected Address alias (either E.164 or Public Party Number). Then, it takes one of the following courses of action depending on circumstances:

- If it finds the Q.931 Connected Number IE, the Oracle Enterprise Session Border Controller extracts the screening indicator and the presentation indicator.
- If there is no Q.931 Connected Number IE, the Oracle Enterprise Session Border Controller extracts the screening indicator and the presentation indicator from the H.225 Connect-UUIE of the Connect message.

With these pieces of information in place, the Oracle Enterprise Session Border Controller performs the conversion from H.323 Connected Number IE to SIP P-Asserted-Identity (PAI) header if and only if the screening indicator is either one of the following:

- Network provided
- User-provided, verified and passed

Then the Oracle Enterprise Session Border Controller adds a SIP PAI header (with URI value) to the 200 OK message that it sends in the SIP call leg. The user part of the URI is set to the value of the Q.931 Connected Number IE's numberDigits field, or to dialDigits value from the Connected Address alias. When the number type is a national number, the Oracle Enterprise Session Border Controller adds a plus sign (+) and the IWF country code (that you configure) to the beginning of the user part. If the number type is an international number, the Oracle Enterprise

Session Border Controller only adds a plus sign (+). And when the Connected Number is empty, the Oracle Enterprise Session Border Controller sets the user part of the PAI header URI to anonymous. When the value in the presentation indicator is Presentation restricted, the Oracle Enterprise Session Border Controller adds the SIP Privacy header (with the value id) to the 200 OK.

In cases when it does not find a screening indicator, the Oracle Enterprise Session Border Controller will not perform the conversion from the H.323 Connected Number IE to the SIP P-Asserted-Identity (PAI) header.

H.323 to SIP Calls

For this type of call, the Oracle Enterprise Session Border Controller checks the 200 OK message for a SIP PAI header and a SIP Privacy header. Before it sends a Connect message on the H.323 call leg, the Oracle Enterprise Session Border Controller generates a Connected Number. It uses the Connected Number to insert a Q.931 Connected Number IE and an H.225 Connected Address alias (type E.164) into the Connect message. The Connected Number is generated in this way:

- If the
  - SIP PAI header is not found, or
  - User part of its URI value is unknown or anonymous, or
  - User part of its URI does not follow the H.225 NumberDigits syntax,

    then the Connect Number that the Oracle Enterprise Session Border Controller generates is a Q.931 Connected Number IE that has no digits and a number type of unknown. In this case, the Oracle Enterprise Session Border Controller will not insert an H.225 Connected Address alias into the Connect message.

    The presentation indicator is set to Number not available due to interworking, and the screening indicator to Network provided. The H.225 NumberDigits's syntax requires that it be between 1 and 128 characters, and only contain these characters: 0 through 9, the pound sign (#), the asterisk (*), and the comma (,).
- In all other cases, the Oracle Enterprise Session Border Controller uses the user part of the URI as the digits for the Connected Number after it performs the following:
  - Strips the plus sign in front of the number, if there is one
  - Strips the IWF country code at the beginning of the number, if there is one

Then the Oracle Enterprise Session Border Controller inserts the Connected Number into the Connect message as the Q.931 Connected Number IE and an H.225 Connected Address alias (type E.164).

If the IWF country code is found in the PAI, the Oracle Enterprise Session Border Controller sets the type of Q.931 Connected Number IE to National Number. Otherwise, the Oracle Enterprise Session Border Controller sets it to international. The screening indicator is set to Network provided, and the presentation indicator is set to Presentation Restricted if the Oracle Enterprise Session Border Controller finds a SIP Privacy header with a value of id, or Presentation Allowed is there is not SIP Privacy header.

IWF COLP COLR Configuration

You configure IWF COLP/COLR support in the IWF configuration by setting two options:

- colp-colr-iwf—Setting this option enables support for IWF COLP/COLR
- colp-colr-country-code—Must be set if you configure the colp-colr-iwf option to recognize or build a national number; the value you enter here:
  - Must be a string of digits from 0 to 9
  - Cannot exceed 32 digits
  - Cannot contain any non-numeric characters; while it allows you to enter them, the system ignores any non-digits characters and so the feature might not work as needed

    To enable IWF COLP/COLR support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the signaling-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type iwf-config and press Enter. The system prompt will change to let you know that you can configure individual

```
ACMEPACKET(session-router)# iwf-config
```

4. options—Set the options parameter by typing options, a Space, the option names with a plus sign in front, and then press Enter.

   Your entry for the colp-colr-country-code option require that you type in the

   entire option name, an equal sign (=), and then the country code value.

   To enter both options at once, separate the two with one command and enclose your entire entry in quotation marks ( ); see the following example for command-line syntax.

```
ACMEPACKET(iwf-config)# options +colp-colr-iwf,colp-colr-country-code=1
```

   If you type this enter without the plus sign, you will overwrite any previously configured options. In order to append options to the IWF configuration's options list, you must prepend the new options with a plus sign as shown in the previous example.

5. Save and activate your configuration.

Options for Calls that Require the IWF

You can configure several specific behaviors by configuring options for calls that require the IWF, and set them for the H.323 side of the call. These options are listed and defined in the table below. Options can be configured either globally for the H.323 configuration, individually for an H.323 interface, or for H.323 session agents.

Global Configuration for H.323

To configure options globally for H.323:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

   From this point, you can configure H.323 parameters. To view see all H.323 parameters, enter a ? at the system prompt.

4. Type options, a space, and the name of the option you want to use. In this example, the MapG729 will map H.245 G.729 to SDP G.729 with Annex B and vice versa.

```
ACMEPACKET(h323)# options MapG729
```

Individual Configuration for H.323

To configure options per individual H.323 interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter.

```
ACMEPACKET(session-router)# h323
```

4. Type h323-stacks and press Enter. The system prompt changes again to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

From this point, you can configure H.323 interface parameters. To view see all H.323 interface parameters, enter a ? at the system prompt.

5. Type options, a space, and the name of the option you want to use. In this example, the MapG729 will map H.245 G.729 to SDP G.729 with Annex B and vice versa.

```
ACMEPACKET(h323-stack)# options
```

Configuring H.323 SA Options

To configure options for H.323 session agents:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
```

From this point, you can configure session agent parameters. To view see all session agent parameters, enter a ? at the system prompt.

4. Type options, a space, and the name of the option you want to use. In this example, the MapG729 will map H.245 G.729 to SDP G.729 with Annex B and vice versa.

```
ACMEPACKET(h323-stack)# options MapG729
```

H.323 SA Options

Options	Description
MapG729	Oracle Enterprise Session Border Controller maps H.245 G.729 to SDP G.729 with Annex B and vice versa. Applicable only to calls that require the IWF.
ColonG729	Oracle Enterprise Session Border Controller uses the : (colon) instead of the = (equal sign) in the media attribute line a=fmtp:18 annexb=yes/no when mapping H.245 G.729 or SDP G.729 with Annex B. Applicable only to calls that require the IWF.
IwfLRQ	Oracle Enterprise Session Border Controller sends an INVITE (with no SDP) to a redirect server in response to an incoming LRQ received on an H.323 interface. If a 3xx message with a redirected contact header is returned, the Oracle Enterprise Session Border Controller will send an LCF in response to the LRQ. Otherwise, it will send an LRJ.
NoG729AnnexB	SDP received by the IWF with H.729 and no FMTP will be mapped to G.729 on the H. 323 side of the call. Can also be set in the session agent options parameter.
sameT38Port	Oracle Enterprise Session Border Controller's H.323 process does not allocate separate ports for audio and T.38. Oracle Enterprise Session Border Controller will send the same audio port in the OLCAck that it sees in a request mode for T.38 and a new OLC for T.38.
pvtStats	Oracle Enterprise Session Border Controller includes program value tree (PVT) statistics in the show h323d display that are a sum of the PVT statistics for all H.323 interfaces. Used for debugging purposes.

Options	Description
acceptAl	Oracle Enterprise Session Border Controller accepts all the codecs received in the SIP 200OK and builds the TCS accordingly.

Suppress SIP Reliable Response Support for IWF

For IWF-originated calls, the Oracle Enterprise Session Border Controller now allows you to configure the suppression of the SIP 100rel option tag on a per-H.323 interface (stack) basis.

When a calls originates on the H.323 side for a call that requires interworking between H.323 and SIP, the Oracle Enterprise Session Border Controller inserts the 100rel option tag in the Supported header of the outgoing SIP INVITE. Although this behavior is required for RFC 3262 conformance, and is ignored by endpoints that do not support this RFC, suppressing the reliable response can alleviate processing burdens and avoid the possibility that an endpoint could mishandle the response.

In addition, enabling this feature suppresses the same 100rel options tag in the Required header for outgoing IWF responses for which an incoming SIP INVITE had that same tag in its Supported header. If an incoming INVITE requires reliable provisional responses and the SIP feature configuration is set to accept the 100rel, the Oracle Enterprise Session Border Controller then includes the 100rel option tag in the outgoing response's Required header. When the SIP feature is not so configured, the Oracle Enterprise Session Border Controller rejects the INVITE with a 420 Bad Extension response.

Without this option, you can suppress the reliable response on a global basis or per SIP next-hop by using the SIP feature configuration. However, using this feature allows a finer degree of granularity by making the functionality only applicable to IWF calls that originate in H.323.

suppress100rel Configuration

To suppress the SIP 100rel option tag:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type h323 and press Enter.

   ```
   ACMEPACKET(session-router)# h323
   ACMEPACKET(h323)#
   ```

4. Type h323-stacks and press Enter.

   ```
   ACMEPACKET(h323)# h323-stacks
   ACMEPACKET(h323-stack)#
   ```

   If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI select command) the configuration that you want to edit.

5. options—Set the options parameter by typing options, a Space, the option name suppress100rel with a plus sign in front of it, and then press Enter.

   ```
   ACMEPACKET(h323-stack)# options +suppress100rel
   ```

   If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new option to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

IWF Codec Negotiation H.323 Slow Start to SIP

For instances when the Oracle Enterprise Session Border Controller is translating a call initiated in H.323 slow start to SIP, you can enable a setting in the IWF configuration that prevents the sending an SDP offer in the SIP INVITE. Instead, the Oracle Enterprise Session Border Controller expects to see an SDP offer from the SIP endpoint in a provisional or reliable/provisional 200 OK, and then sends an answer in an ACK or PRACK.

With this parameter disabled (default), the Oracle Enterprise Session Border Controller populates the SIP INVITE with SDP based on the media profiles applied to the ingress H.323 session agent or the IWF configuration.

IWF Codec Negotiation Configuration

To prevent the Oracle Enterprise Session Border Controller from sending an SDP offer in the SIP INVITE for a call being translated between H.323 slow start and SIP:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type iwf-config and press Enter.

```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```

4. slow-start-no-sdp-in-invite—Enable this parameter if you want to prevent the Oracle Enterprise Session Border Controller from sending an SDP offer in the SIP INVITE for an IWF call initiated in H.323 slow start (being translated to SIP). The default is disabled. Valid values are:
   - enabled | disabled
5. Save and activate your configuration.

IWF H.245 Signaling Support for G.726

In addition to providing G.726 support for pure SIP and pure H.323 calls, the Oracle Enterprise Session Border Controller supports the G.726 payload type for H.245 and calls that require interworking (IWF) between SIP and H.323.

For IWF calls using ITU-T G.726 as the audio codec, the SIP call leg requires G.726 in the SDP. The H.323 side of the call signals G.726 (in the H.245 openLogicalChannel and TerminalCapabilitySet messages) by including a GenericCapability defining G.726 as the codec. In the GenericCapability, the capabilityIdentifier and maxBitRate parameters identify G.726. While a capabilityIdentifier with 0.0.7.726.1.0 designates G.726, the maxBitRate designate the data transmission rate.

Codec	Max Bit Rate	Data Rate
G726-16	160	16 kbit/s
G726-24	240	24 kbit/s
G726-32	320	32 kbit/s
G726-40	400	40 kbit/s

To support G.726 for IWF calls, the Oracle Enterprise Session Border Controller converts the G726-X value in the SDP of SIP messages to a GenericCapability structure in H.323/H.245 messages, and the conversion works the same way in reverse.

H.245 and G.726 Configuration

To enable this feature, you do need to set up media profile configurations appropriately. Media profiles now allow you to set the configuration to any of the four G.726 encodings (as defined by ITU G726 Annex B and RFC 3551). You must create one media profile for each of the four different supported data rates. In addition, you are also required to set a genericAudioCapability media profile.

Media Profile for H.245 and G.726 Configuration

To set a media profile for H.245 and IWF G.726 support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. name—Set the name of the media profile to G726-16. Values to support this feature are: G726-16, G726-24, G726-32, and G726-40.
5. media-type—Set the media type to use for this media profile; for generic video, set this parameter to audio. Valid values are:
   - audio | video | application | data
6. payload-type—Set the payload type to use for the generic video media profile.
7. transport—Set the transport type to use for the generic video media profile. The default value is RTP/AVP. Valid values are:
   - UDP | RTP/AVP
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

   The following is a sample of a media profile configuration for H.245/IWF G.726 support:

```
media-profile
        name                            g726-40
        media-type                      audio
        payload-type                    105
        transport                       RTP/AVP
        req-bandwidth                   0
        frames-per-packet               0
        parameters
        average-rate-limit              0
        sdp-rate-limit-headroom         0
        sdp-bandwidth                   disabled
```

Media Profile Configuration for Generic Audio Support

To set a media profile for generic audio support:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. name—Set the name of the generic audio media profile to genericAudioCapability. There is no default for this parameter.
5. media-type—Set the media type to use for this media profile; for generic video, set this parameter to audio. Valid values are:
   - audio | video | application | data
6. payload-type—Set the payload type to use for the generic audio media profile.
7. transport—Set the transport type to use for the generic audio media profile. The default value is RTP/AVP. Valid values are:
   - UDP | RTP/AVP
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

   The following is a sample of a generic audio media profile configuration:

```
media-profile
        name                            genericAudioCapability
        media-type                      audio
        payload-type                    104
        transport                       RTP/AVP
        req-bandwidth                   0
        frames-per-packet               0
        parameters
        average-rate-limit              0
        sdp-rate-limit-headroom         0
        sdp-bandwidth                   disabled
```

Flow Control Mapping for Interworking Function (IWF) Video

H.245 is a protocol for the transmission of call management and control signals in networks using H.323 equipment. The H.245 specification is used in audio, video, and data transmissions, as well as in voice over IP (VoIP). H.245 messages are sent over special channels called H.245 control channels.

H.245 signaling is used to manage and control call setup and connection. Functions of H.245 include determining which endpoint is to be the master and which is to be the slave during the call, opening and closing of multiplexed data-transfer paths between the endpoints, establishing an upper limit to the data transfer speed on each logical channel, information exchanges between endpoints concerning the types of data each endpoint can send and receive, requests by the receiving endpoint for changes in the mode of the data sent by the transmitting endpoint, and requests by either endpoint to end the call.

In the H.245 standard, the FlowControlCommand message is used to specify the upper limit of bit rate of either a single logical channel or the whole multiplex. The following is an excerpt from the H.245 standard.

Command Message: Flow Control (from H.245 standard)

```
==================================================
FlowControlCommand ::= SEQUENCE
{
        scope CHOICE
    {
    logicalChannelNumber LogicalChannelNumber,
    resourceID INTEGER (0..65535),
    wholeMultiplex NULL
    },
    restriction CHOICE
    {
    maximumBitRate INTEGER (0..16777215), -- units 100 bit/s
    noRestriction NULL
```

```
    },
    ...
}
==================================================
```

A terminal may send this command to restrict the bit rate that the far-end terminal sends. A receiving terminal must comply with this command.

In an H.323 environment, the Oracle Enterprise Session Border Controller previously used the FlowControlCommand to map to SIP using either the Real-Time Control Protocol (RTCP) feedback function, or the SIP signaling path (for example, the INFO method).

The Oracle Enterprise Session Border Controller now supports the SIP counter part of the H.245 FlowControlCommand using the SIP signaling path with the INFO method. The Oracle Enterprise Session Border Controller sends the SIP INFO message with "change_bitrate" rate parameter that has the value 100* maxBitRate from the corresponding H.245 FlowControlCommnad message. For example, in the following messages, the incoming H.323 message with the H.245 FlowControlCommand, is converted into the outgoing SIP INFO message with the message body.

Incoming H.323 Message with H.245 FlowControlCommand:

```
H.245
PDU Type: command (2)
        command: flowControlCommand (4)
            flowControlCommand
                scope: logicalChannelNumber (0)
                    logicalChannelNumber: 102
                restriction: maximumBitRate (0)
                    maximumBitRate: 4480
```

Outgoing SIP INFO Message:

```
Message Body
        eXtensible Markup Language
            <?xml
                version="1.0"
                encoding="utf-8"
                ?>
            <media_control>
                <vc_primitive>
                    <to_encoder>
                        <change_bitrate>
                            4480000
                            </change_bitrate>
                        </to_encoder>
                    </vc_primitive>
                </media_control>
```

Customized G.729 Support

The Oracle Enterprise Session Border Controller supports the use of custom G.729 encoding for calls that require interworking between SIP and H.323. If you use a proprietary G.729 encoding format in your network, then you might need to use this feature.

When you set the acceptG729abFormat option in the global H.323 configuration, the Oracle Enterprise Session Border Controller performs conversions like those in the following examples:

- For calls initiated in SIP, the Oracle Enterprise Session Border Controller can parse RTP map strings such as G.729a and G.729ab in the SDP, and then map them to H.245 data types.
  - G.729a becomes g729AnnexA.
  - G.729ab becomes g729AnnexAwAnnexB.

- For calls initiated in H.323, the Oracle Enterprise Session Border Controller can create non-standard RTP map strings such as G.729a and G.729ab from mapped H.245 data types.
  - g729 becomes G729.
  - g729AnnexA becomes G.729a.
  - g729AnnexAwAnnexB becomes G.729ab.

When you enable the acceptG729abFormat option, the Oracle Enterprise Session Border Controller performs customized G.729 mapping in the following instances.

- For calls initiated in SIP and translated to H.323, the Oracle Enterprise Session Border Controller:
  - Converts the SDP in an incoming SIP INVITE to a list of fastStart OpenLogicalChannel requests that are in turn included in the outgoing Setup message.
  - Converts the list of fastStart OpenLogicalChannelAck responses (which can be received in any message up to and including the Connect message) to SDP sent with a SIP response.
- For calls initiated in H.323 and translated to SIP, the Oracle Enterprise Session Border Controller:
  - Converts the list of fastStart OpenLogicalChannel requests to SDP in the outgoing SIP INVITE.
  - Converts SDP in a SIP response (such as a 200 OK) to the list of fastStart OpenLogicalChannelAck responses included with the callProceeding, Progress, Alerting, or Connect message. This depends on when the SDP is received on the SIP side.
- For all IWF calls regardless of initiating protocol, the Oracle Enterprise Session Border Controller:
  - Converts SDP on the SIP side to the terminalCapabilitySet message to be sent on the H.323 side.

Also note that when the format is G729, the Oracle Enterprise Session Border Controller maps it to g729wAnnexB if the a=fmtp:18 annexb=yes attribute is present. When the a=fmtp:18 annexb=no attribute is present, the Oracle Enterprise Session Border Controller maps G729 to g729. And with no a=fmtp:18 annexb=no attribute, the Oracle Enterprise Session Border Controller also maps G729 to g729 when this option is enabled.

The Oracle Enterprise Session Border Controller also maps G729 to g729 because pure G729 with static payload type 18 does not include an fmtp attribute where annexb=no.

About Dynamic Payload Mapping

G.729a and G.729ab use dynamic payload types, but the Oracle Enterprise Session Border Controllerdoes not propagate these dynamic payload types to corresponding dynamicRTPPayloadType (an optional field in OpenLogicalChannel requests) on the H.323 side.

For an IWF call initiated in H.323, the dynamic payload types for G.729a and G.729ab are retrieved from media profile configurations when the Oracle Enterprise Session Border Controller converts the list of fastStart OpenLogicalChannel requests to SDP sent on the SIP side. As a result, you must set up media profile configurations for G.729a and G.729ab for the feature to work properly. In these media profiles, the following parameters must be set as follows:

- name—For the G.729a profile, set the name to G.729a. For the G.729ab profile, set the name to G.729ab.
- payload-type—For each media profile (G.729a and G.729ab), DO NOT use payload type 18, which is the static payload type used for G729.

Customized G.729 Configuration

This section shows you how to configure the acceptG729abFormat option in the global H.323 configuration.

To enable customized G.729 support for IWF calls:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323-config and press Enter.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

If you are adding this feature to a pre-existing configuration, select the configuration to edit it.

4. options—Set the options parameter by typing options, a Space, the option name acceptG729abFormat with a plus sign in front of it. Then press Enter.

```
ACMEPACKET(h323-stack)# options +acceptG729abFormat
```

If you type options and then the option value for either of these entries without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save and activate your configuration.

SIP-H.323 IWF Support for H.264 and H.263+

Signaling protocol interworking between SIP and H.323 supports the H.264 and H.263+ video codecs.

H.264 in H.323 (H.241)

This section describes the H.264 capabilities and media packetization in H.323. Capability exchange signaling looks like this:

```
openLogicalChannel .  SEQUENCE [EMPTY -1] ...
forwardLogicalChannelNumber = 3 .  INTEGER [EMPTY -1] (1..65535)
forwardLogicalChannelParameters .  SEQUENCE [EMPTY -1] ...
 ..   dataType .  CHOICE [EMPTY -1] ...
. . . . . videoData .  CHOICE [EMPTY -1] ...
. . . . . genericVideoCapability .  SEQUENCE [EMPTY -1] ...
. . . . . capabilityIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 7 {itu-t recommendation h 241 0 0 1}.OBJECT
IDENTIFIER [EMPTY-1]
. . . . . maxBitRate = 4480 .  INTEGER [EMPTY -1] (0..-1)
. . . . . collapsing .  SEQUENCE OF [EMPTY -1] SEQUENCE [EMPTY -1] ...
. . . . . * .  SEQUENCE [EMPTY -1] ...
. . . . . parameterIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 41 .  INTEGER [EMPTY -1] (0..127)
. . . . . parameterValue .  CHOICE [EMPTY -1] ...
. . . . . booleanArray = 64 .  INTEGER [EMPTY -1] (0..255)
. . . . . * .  SEQUENCE [EMPTY -1] ...
. . . . . parameterIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 42 .  INTEGER [EMPTY -1] (0..127)
. . . . . parameterValue .  CHOICE [EMPTY -1] ...
. . . . . unsignedMin = 29 .  INTEGER [EMPTY -1] (0..65535)
. . . . . * .  SEQUENCE [EMPTY -1] ...
. . . . . parameterIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 3 .  INTEGER [EMPTY -1] (0..127)
. . . . . parameterValue .  CHOICE [EMPTY -1] ...
. . . . . unsignedMin = 81 .  INTEGER [EMPTY -1] (0..65535)
. . . . . * .  SEQUENCE [EMPTY -1] ...
. . . . . parameterIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 6 .  INTEGER [EMPTY -1] (0..127)
. . . . . parameterValue .  CHOICE [EMPTY -1] ...
. . . . . unsignedMin = 15 .  INTEGER [EMPTY -1] (0..65535)
. . . . . * .  SEQUENCE [EMPTY -1] ...
. . . . . parameterIdentifier .  CHOICE [EMPTY -1] ...
. . . . . standard = 4 .  INTEGER [EMPTY -1] (0..127)
. . . . . parameterValue .  CHOICE [EMPTY -1] ...
```

```
. . . . . . . . . . unsignedMin = 7 .  INTEGER [EMPTY -1] (0..65535)
. . . . multiplexParameters .  CHOICE [EMPTY -1] ...
. . . . . h2250LogicalChannelParameters .  SEQUENCE [EMPTY -1] ...
. . . . . . sessionID = 2 .  INTEGER [EMPTY -1] (0..255)
. . . . . . mediaControlChannel .  CHOICE [EMPTY -1] ...
. . . . . . . unicastAddress .  CHOICE [EMPTY -1] ...
. . . . . . . . iPAddress .  SEQUENCE [EMPTY -1] ...
. . . . . . . . . network = 4 '.e.' =0xac10650b <172.16.101.11> .OCTET STRING
[EMPTY -1]
. . . . . . . . . tsapIdentifier = 50137 .  INTEGER [EMPTY -1] (0..65535)
. . . . . . dynamicRTPPayloadType = 109 .  INTEGER [EMPTY -1] (96..127)
. . . . . . mediaPacketization .  CHOICE [EMPTY -1] ...
. . . . . . . rtpPayloadType .  SEQUENCE [EMPTY -1] ...
. . . . . . . . payloadDescriptor .  CHOICE [EMPTY -1] ...
. . . . . . . . . oid = 8 {itu-t recommendation h 241 0 0 0 0}.OBJECT
IDENTIFIER [EMPTY -1]
. . . . . . . . payloadType = 109 .  INTEGER [EMPTY -1] (0..127)
```

This table outlines H.241 to H.264 mappings.

Identifier	Description
Capability name	ITU-T Rec H.241 H.264 Video Capabilities
Capability identifier type	Standard
Capability identifier value	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) generic-capabilities(1)}
maxBitRate	This field shall be included, in units of 100 bit/s. This field represents the maximum bitrate of the H.264 Type II bitstream as defined in Annex C/H.264.
collapsing	This field shall contain the H.264 Capability Parameters as given below.

Capabilities

The H.264 capability set is structured as a list of one or more H.264 capabilities, each of which has:

- Profile (mandatory)
- Level (mandatory)
- Zero or more additional parameters

These capabilities communicate the ability to decode using one or more H.264 profiles contained in a GenericCapability structure. For each H.264 capability, optional parameters can appear. These parameters permits a terminal to communicate that it has capabilities in addition to meeting the support requirements for the signaled profile and level.

Optional parameters include: CustomMaxMBPS, CustomMaxDPB, CustomMaxBRandCPB, MaxStaticMBPS, max-rcmd-unit-size, max-nal-unit-size, SampleAspectRatiosSupported, AdditionalModesSupported, amd AdditionalDisplayCapabilities.

H.264 Media Packetization

For H.323, systems signal their H.264 mediaPacketization by including: MediaPacketizationCapability.rtpPayload.Type.payloadDescriptor.oid, with the OID having the value {itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) h241AnnexA(0)}.

In compliance with RFC 3984's non-interleaved mode, the following is supported: MediaPacketizationCapability.rtpPayloadType.payloadDescriptor.oid, with the OID having the value {itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) RFC3984NonInterleaved(1)}.

In compliance with RFC 3984's interleaved mode, the following is supported: MediaPacketizationCapability.rtpPayloadType.payloadDescriptor.oid, with the OID having the value {itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0)iPpacketization(0) RFC3984Interleaved(2)}.

H.264 in SIP

H.264 in SIP can contain these optional parameters, which be included in the "a=fmtp" line of SDP if they appear: profile-level-id, max-mbps, max-fs, max-cpb, max-dpb, maxbr, redundant-pic-cap, sprop-parameter-sets, parameter-add, packetization-mode, spropinterleaving-depth, deint-buf-cap, sprop-deint-buf-req, sprop-init-buf-time, sprop-max-dondiff, and max-rcmd-nalu- size.

The profile-level-id parameter is a base 16[6] hexidecimal representation of the following three bytes in sequence:

1. profile_idc
2. profile_oip—Composed of the values from constraint_set0_flag, constraint_set1_flag, constraint_set2_flag, and reserved_zero_5bits—in order of bit significance, starting from the most significant bit.
3. level_idc—Note that reserved_zero_5bits is required to be equal to 0 in [1], but other values for it may be specified in the future by ITU-T or ISO/IEC.

H.264 Packetization Mode

In SIP, the packetization-mode parameter signals the properties of the RTP payload type or the capabilities of a receiver's implementation. Only a single configuration point can be indicated. So when capabilities support more than one packetization-mode are declared, multiple configuration points (RTP payload types) must be used.

- When the value of packetization-mode equals 0 or packetization-mode is not present, the single NAL mode is used.
- When the value of packetization-mode equals 1, the non- interleaved mode is used.
- When the value of packetization-mode equals 2, the interleaved mode is used.

This example shows a SIP offer-answer exchange. Here is the offer SDP:

```
m=video 49170 RTP/AVP 100 99 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; packetization-mode=0;
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
```

And here is the answer SDP for the example:

```
m=video 49170 RTP/AVP 100 99 97
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42A01E; packetization-mode=0;
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
```

H.264 IWF Conversions

This section contains two table that show profile, level, and media packetization conversions for H.264 undergoing interworking.

Profile	H.264 in SIP	H.264 (H.241 in H.323)
H264_PROFILE_STR_BASELINE	66	64
H264_PROFILE_STR_MAIN	77	32
H264_PROFILE_STR_EXTENDED	88	32

H.264 Level	H.2264 in SIP	H.264 (H.241 in H.323)	Constraints
1	10	15	0x00
1b	11	19	0x10
1.1	11	22	0x00
1.2	12	29	0x00
1.3	13	36	0x00
2	20	43	0x00
2.1	21	50	0x00
2.2	22	57	0x00
3	30	64	0x00
3.1	31	71	0x00
3.2	32	78	0x00
4	40	85	0x00
4.1	41	92	0x00
4.2	42	99	0x00
5	50	106	0x00
5.1	51	113	0x00

H.264 SIP Packetization	H.264 (H.241 in H.323) OID in mediaPacketization
packetization-mode=0	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) h241AnnexA(0)}
packetization-mode=1	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) RFC3984NonInterleaved(1)}
packetization-mode=2	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPpacketization(0) RFC3984Interleaved(2)}

IWF Unsupported Parameters

The following H.241 parameters are not supported for interworking: CustomMaxMBPS, CustomMaxFS CustomMaxDPB, CustomMaxBRandCPB, MaxStaticMBPS, max-rcmd-nal-unit-size, max-nal-unit-size, SampleAspectRatiosSupported, AdditionalModesSupported, and AdditionalDisplayCapabilities.

The following SDP parameters are not supported for interworking: max-mbps, max-fs, max-cpb, max-dpb, maxbr, redundant-pic-cap, sprop-parameter-sets, parameter-add, spropinterleaving-depth, deint-buf-cap, sprop-deint-buf-req, sprop-init-buf-time, sprop-max-dondiff, and max-rcmd-nalu-size.

H.263+ in H.323

This section describes the H.264 capabilities and media packetization in H.323. Capability exchange signaling looks like this:

```
. . . . . capability .  CHOICE [EMPTY -1] ...
. . . . . . receiveVideoCapability .  CHOICE [EMPTY -1] ...
. . . . . . . h263VideoCapability .  SEQUENCE [EMPTY -1] ...
. . . . . . . . sqcifMPI = 1 .  INTEGER [EMPTY -1] (1..32)
. . . . . . . . qcifMPI = 1 .  INTEGER [EMPTY -1] (1..32)
. . . . . . . . cifMPI = 1 .  INTEGER [EMPTY -1] (1..32)
. . . . . . . . maxBitRate = 1000 .  INTEGER [EMPTY -1] (1..192400)
. . . . . . . . unrestrictedVector = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . arithmeticCoding = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . advancedPrediction = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . pbFrames = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . temporalSpatialTradeOffCapability = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . errorCompensation = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . h263Options .  SEQUENCE [EMPTY -1] ...
. . . . . . . . . advancedIntraCodingMode = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . deblockingFilterMode = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . improvedPBFramesMode = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . unlimitedMotionVectors = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . fullPictureFreeze = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . partialPictureFreezeAndRelease = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . resizingPartPicFreezeAndRelease = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . fullPictureSnapshot = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . partialPictureSnapshot = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . videoSegmentTagging = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . progressiveRefinement = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . dynamicPictureResizingByFour = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . dynamicPictureResizingSixteenthPel = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . dynamicWarpingHalfPel = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . dynamicWarpingSixteenthPel = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . independentSegmentDecoding = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . slicesInOrder-NonRect = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . slicesInOrder-Rect = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . slicesNoOrder-NonRect = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . slicesNoOrder-Rect = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . alternateInterVLCMode = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . modifiedQuantizationMode = 1 .  BOOLEAN [EMPTY -1]
. . . . . . . . . reducedResolutionUpdate = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . separateVideoBackChannel = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . videoBadMBsCap = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . h263Version3Options .  SEQUENCE [EMPTY -1] ...
. . . . . . . . . . dataPartitionedSlices = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . fixedPointIDCT0 = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . interlacedFields = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . currentPictureHeaderRepetition = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . previousPictureHeaderRepetition = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . nextPictureHeaderRepetition = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . pictureNumber = 0 .  BOOLEAN [EMPTY -1]
. . . . . . . . . . spareReferencePictures = 0 .  BOOLEAN [EMPTY -1]
```

H.263+ in SIP

H.263+ in SIP appears looks like this:

```
a=rtpmap:100 H263-1998/90000
a=fmtp:100 CIF=1; QCIF=1; SQCIF=1; D=1; F=1; I=1; J=1; L=1; S=1; T=1
a=rtpmap:34 H263/90000
a=fmtp:34 CIF=1; QCIF=1; SQCIF=1
```

H.263+ IWF Conversions

This section contains a table showing H.263+ conversions for SIP-h.323 interworking.

H.263+ in H.323 Parameters (Annex) in ftmp line	H.263+ in SIP
sqcifMPI	SQCIF
qcifMPI	QCIF
cifMPI	CIF
	CIF4
	CIF16
maxBitRate	
unrestrictedVector	D
arithmeticCoding	E
advancedPrediction	F
pbFrames	G
temporalSpatialTradeOffCapability	
errorCompensation	H
h263Options	
advancedIntraCodingMode	I
deblockingFilterMode	J
improvedPBFramesMode	
unlimitedMotionVectors	
fullPictureFreeze	L
partialPictureFreezeAndRelease	
resizingPartPicFreezeAndRelease	
fullPictureSnapshot	
partialPictureSnapshot	
videoSegmentTagging	
progressiveRefinement	
dynamicPictureResizingByFour	P = 1
dynamicPictureResizingSixteenthPel	P = 2
dynamicWarpingHalfPel	P = 3
DynamicWarpingSixteenthPel	P = 4
independentSegmentDecoding	R
slicesInOrder-NonRect	K = 1
slicesInOrder-Rect	K = 2
slicesNoOrder-NonRect	K = 3
slicesNoOrder-Rect	K = 4

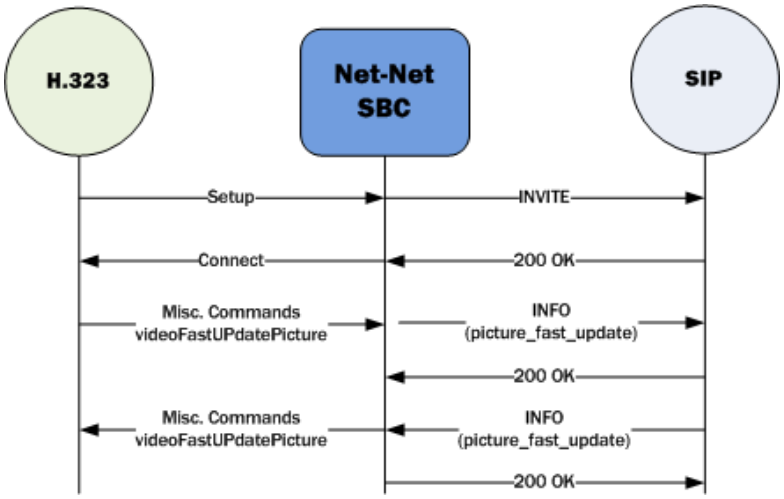
H.263+ in H.323 Parameters (Annex) in ftmp line	H.263+ in SIP
alternateInterVLCMode	S
modifiedQuantizationMode	T
reducedResolutionUpdate	Q
separateVideoBackChannel	
videoBadMBsCap	
	PAR
	CPCF
	CUSTOM
h263Version3Options	

IWF Unsupported Parameters

The following optional SDP parameters are not supported for H.263+ interworking: SQCIF, QCIF, CIF, CIF4, CIF16, CUSTOM, PAR, CPCF.

SIP-H.323 IWF in Video Conferencing Applications

For video conferencing and other video applications, the Oracle Enterprise Session Border Controller supports interworking between the H.323 Miscellaneous Commands videoFastUpdatePicture and the SIP INFO containing XML schema for Full Update. The noted H.323 message commands the video encoder to enter fast-update mode.



There is no configuration required for the interworking between these two messages to work.

International Peering with IWF and H.323 Calls

When you do not enable this feature, SIP to H.323 IWF calls default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option cpnType=X, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outing calls on the h323-stack.

The following is a list of possible cpnType=X option values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

International Peering Configuration

You configure this feature as an option in the h323-stack configuration.

To configure the cpnType=X option for H323-H323 calls:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323)#
```

4. Type h323-stacks and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

5. Set the options parameter by typing options, a Space, the option name cpnType=x with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(h323-stack)# options +cpnType=x
```

   If you type options without the plus sign, you will overwrite any previously configured options. In order to append the new options to the h323-stack's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Save and activate your configuration.

IWF Codec Renegotiation for Audio Sessions

For calls requiring interworking between SIP and H.323, there can be several instances for audio sessions when a mid-call codec change is necessary. These are some examples of when the codec used for voice transportation is necessary:

- Sessions between analog FAX machines that start as regular voice calls but then must use a codec that is fax-signalling tolerant (like transparent G.711) when FAX tones are detected; detection takes place after the call has been answered. The case of modem calls is similar.
- An established call is redirected in one carrier's network either to a different enduser or to a media server. In this case, the party to which the call is redirected might not support the codec used in the redirection. If request for a codec change is carried out at the signalling level, the call can proceed with the party to which the call was redirected.
- Endusers might want to change codecs when they suffer low voice quality.

Both SIP and H.323 provide mechanisms for changing codecs during a call: SIP uses the ReINVITE, and H.323 uses the H.245 Request Mode. Using the option called processRequestModeForIWF=all either in an H.323 interface (stack) or an H.323 session agent configuration, you can enable the Oracle Enterprise Session Border Controller to interwork SIP ReINVITE and H.245 Request Mode requests.

RTN 1976

Codec Request Change from the SIP Side

When a SIP party requests a code change, the Oracle Enterprise Session Border Controller communicates with the H.323 endpoint to renegotiate support for an updated codec. In this renegotiation, the Oracle Enterprise Session Border Controller presents codec for use ordered according to the SIP side's preference and one is selected. Then the Oracle Enterprise Session Border Controller handles opening of a new logical channel that uses the updated codec, and closes the old logical channel (that uses the now-outdated codec). On the SIP side, the Oracle Enterprise Session Border Controller sends a 200 OK with the necessary RTP port and codec information for the new logical channel.

Codec Request Change from the H.323 Side

When the Oracle Enterprise Session Border Controller receives a codec request change on the H.323 side of an IWF call, it sends a Re-INVITE to the SIP endpoint containing new codec and information. The Oracle Enterprise Session Border Controller uses IP address and port information it has cached for the H.323 side of the call for the Re-INVITE since H.245 Request Mode requests do not have this data. If the IP address and port combination should subsequently change (in an OLC from the H.323 side), the Oracle Enterprise Session Border Controller handles additional INTVITEs on the SIP side to support the change.

Exceptional Cases

When the relevant option in enabled, the Oracle Enterprise Session Border Controller can handle properly the following cases of codec change:

- When the H.323 side rejects the request mode change, the Oracle Enterprise Session Border Controller response to the SIP side with a 488 Not Acceptable. Session description and state remain unchanged, and the call continues using the original session description.
- When the H.323 side does not respond to the request mode change within the timeout limitation, the Oracle Enterprise Session Border Controller releases the call on both sides.
- When the SIP side does not respond to the ReINVITE within in the timeout limitation, the Oracle Enterprise Session Border Controller releases the call on both sides.
- When the intersection of codec is empty, the Oracle Enterprise Session Border Controller rejects the codec change on the SIP side with a 488 Not Acceptable and on the H.323 side with an H.245 RequestModeReject. Session description and state remain unchanged, and the call continues using the original session description.
- If the Oracle Enterprise Session Border Controller does not receive any of the LogicalChannel request or acknowledgement messages, the Oracle Enterprise Session Border Controller releases the call on both sides.

Note that for protocol timeout errors, the preferred behavior is to release the call on both sides. Timeout errors usually indicate network problems, such as an endpoint being unreachable.

IWF Codec Renogotiation Configuration

You can apply the processRequestModeForIWF=all to H.323 interfaces (stacks) and to H.323 session agents (sessions agents for which H.323 has been identified in the protocol parameter). The example below shows you how to enable this option for an H.323 session agent.

To enable IWF codec renegotiation for an H.323 session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. options—Set the options parameter by typing options, a Space, the option name processRequestModeForIWF=all with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(session-agent)# options +processRequestModeForIWF=all
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save your work.

13

Session Routing and Load Balancing

This chapter explains how to configure session routing and load balancing for SIP and H.323 services. It contains information about configuring session agents and session agent groups, as well as local policies that can be used for routing SIP or H.323 signals.

Routing Overview

This section provides an overview of routing SIP and H.323 sessions when using the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller chooses the next hop through the network for each SIP and H.323 session based on information received from routing policies and constraints. Routing policies can be as simple as routing all traffic to a proxy or routing all traffic from one network to another. Routing policies can also be more detailed, using constraints to manage the volume and rate of traffic that can be routed to a specific network. For example, you can manage volume and rate of traffic to enable theOracle Enterprise Session Border Controller to load balance and route around softswitch failures.

When a call request arrives at the Oracle Enterprise Session Border Controller, a decision making process then occurs to determine whether the message is coming from a session agent. If so, the Oracle Enterprise Session Border Controller checks whether that session agent is authorized to make the call. Local policy is then checked to determine where to send the message on to.

Session Agents Session Groups and Local Policy

When you configure session routing for SIP and H.323, you can use session agents, session agent groups and local policies to define routing. (Using session agents and session agent groups is not required.)

- session agent: defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes.
- session agent group (SAG): contains individual session agents. Members of a SAG are logically equivalent (although they might vary in their individual constraints) and can be used interchangeably.

  You apply an allocation strategy to the SAG to allocate traffic across the group members. Session agent groups also assist in load balancing among session agents.
- local policy: indicates where session request messages, such as SIP INVITES, are routed and/or forwarded. You use a local policy to set a preference for selecting one route over another.

Another element of routing is the realm. Realms are used when a any log level you set here overrides the log level you set in the system configuration’s process log level parameter, communicates with multiple network elements over

a shared intermediate connection. Defining realms allows sessions to go through a connection point between the two networks.

When you configure a realm, you give it an identifier, which stores the name of the realm associated with the any log level you set here overrides the log level you set in the system configuration's process log level parameter.. The realm identifier value is also needed when you configure session agents and local policies. You can associate a realm with a session agent to identify the realm for sessions coming from or going to the session agent. You also need the realm identifier when you configure local policy to identify the egress realm (realm of the next hop).

About Session Agents

This section describes session agents. A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. Service elements such as gateways, softswitches, and gatekeepers are defined automatically within the Oracle Enterprise Session Border Controller as session agents. For each session agent, concurrent session capacity and rate attributes can be defined. You can group session agents together into session agent groups and apply allocation strategies to achieve traffic load balancing.

You can assign a media profile to a session agent and indicate whether the transport protocol is SIP or H.323. If the protocol is H.323, you need to indicate whether the session agent is a gateway or a gatekeeper.

You can configure a set of attributes and constraints for each session agent to support the following:

- session access control: Oracle Enterprise Session Border Controller only accepts requests from configured session agents
- session admission control (concurrent sessions): Oracle Enterprise Session Border Controller limits the number of concurrent inbound and outbound sessions for any known service element.
- session agent load balancing: session agents are loaded based on their capacity and the allocation strategy specified in the session agent group.
- session (call) gapping: Oracle Enterprise Session Border Controller polices the rate of session attempts to send to and receive from a specific session agent.

SIP Session Agents

SIP session agents can include the following:

- softswitches
- SIP proxies
- application servers
- SIP gateways
- SIP endpoints

In addition to functioning as a single logical next hop for a signaling message (for example, where a SIP INVITE is forwarded), session agents can provide information about next or previous hops for packets in a SIP agent, including providing a list of equivalent next hops.

You can use the session agent to describe one or more SIP next or previous hops. Through the configured carriers list, you can identify the preferred carriers to use for traffic coming from the session agent. This set of carriers will be matched against the local policy for requests coming from the session agent. You can also set constraints for specific hops.

Session Agent Status Based on SIP Response

The Oracle Enterprise Session Border Controller can take session agents out of service based on SIP response codes that you configure, and you can also configure SIP response codes that will keep the session agent in service.

With this feature disabled, the Oracle Enterprise Session Border Controller determines session agents' health by sending them ping messages using a SIP method that you configure. Commonly, the method is an OPTIONS request.

If it receives any response from the session agent, then the Oracle Enterprise Session Border Controller deems that session agent available for use.

However, issues can arise when session agents are administratively out of service, but able to respond to OPTIONs requests. A session agent like this might only respond with a 200 OK when in service, and send a 4xx or 5xx message otherwise.

The session agent status feature lets you set the SIP response message that either takes a session agent out of service or allows it to remain in service when it responds to the Oracle Enterprise Session Border Controller's ping request.

Details of this feature are as follows:

- The Oracle Enterprise Session Border Controller only considers a session agent in service when it responds to a request method you set with the final response code that you also set. If a final response code is set, then provisional responses are not used for determining whether or not to take a session agent out of service. If the Oracle Enterprise Session Border Controller receives a final response code that does not match the session agent configuration, it treats the session agent as though it had not responded.
- The Oracle Enterprise Session Border Controller takes a session agent out of service when it receives an error response for dialog creating request with a response code listed in the new out-service-response-codes parameter.

In the case where a the session agent's response has a Retry-After header, the Oracle Enterprise Session Border Controller tries to bring the session agent back into service after the period of time specified in the header. To do so, it sends another ping request.

There are two lists you can configure in the session agent configuration to determine status:

- In-service list—Set in the ACLI ping-in-service-response-codes parameter, this list defines the response codes that keep a session agent in service when they appear in its response to the Oracle Enterprise Session Border Controller's ping request. Furthermore, the Oracle Enterprise Session Border Controller takes the session agent out of service should a response code be used that does not appear on this list.
- Out-of-service list—Set in the ACLI out-service-response-codes parameter, this list defines the response codes that take a session agent out of service when they appear in its response to the Oracle Enterprise Session Border Controller's ping request or any dialog-creating request.

When the Oracle Enterprise Session Border Controller receives a session agent's response to its ping request, it first checks to see if there is an in-service list of responses configured for that session agent. If the list is configured and the Oracle Enterprise Session Border Controller determines that there is a match, the session agent is deemed in service. Otherwise it takes the session agent out of service. In this way, the in-service list takes precedence over the out-of-service list. If you configure the in-service list, then the Oracle Enterprise Session Border Controller ignores the out-of-service list.

If there is no list of in-service responses for the session agent, then the Oracle Enterprise Session Border Controller checks the out of service list. If it is configured and the Oracle Enterprise Session Border Controller determines that there is a match, the Oracle Enterprise Session Border Controller removes that session agent from service. If there is no match, then the session agent is deemed in service.

SIP Session Agent Continuous Ping

You can configure the Oracle Enterprise Session Border Controller to use either a keep-alive or continuous method for pinging SIP session agents to determine their health—i.e., whether or not the Oracle Enterprise Session Border Controller should route requests to them. To summarize the two methods:

- keep-alive— Oracle Enterprise Session Border Controller sends a ping message of a type you configure to the session agent in the absence of regular traffic. Available in Release C5.1.0 and in earlier releases.
- continuous—The Oracle Enterprise Session Border Controller sends a ping message regardless of traffic state (regular or irregular); the Oracle Enterprise Session Border Controller regularly sends a ping sent based on the configured ping interval timer. Available in Release C5.1.1p6 and in later releases.

By sending ping messages, the Oracle Enterprise Session Border Controller monitors session agents' health and can determine whether or not to take a session out of service (OOS), leave it in service, or bring it back into service after being OOS.

When you set it to use the keep-alive mode of pinging (available in Release C5.1.0 and before), the Oracle Enterprise Session Border Controller starts sending a configured ping message to a session agent when traffic for that session agent has become irregular. The Oracle Enterprise Session Border Controller only sends the ping if there are no SIP transactions with a session agent over a configurable period of time, to which the session agent's response can have one of the following results:

- Successful response—A successful response is either any SIP response code or any response code not found in the out-service-response-codes parameter; these leave the session agent in service. In addition, any successful response or any response in the ping-in-service-response-codes parameter can bring a session agent from OOS to in-service status.
- Unsuccessful response—An unsuccessful response is any SIP response code configured in the out-service-response-codes parameter and takes the session agent sending it OOS. Because this parameter is blank by default, the Oracle Enterprise Session Border Controller considers any SIP response code successful.
- Transaction timeout—A transaction timeout happens when the session agent fails to send a response to the Oracle Enterprise Session Border Controller's request, resulting in the session agent's being taken OOS.

Despite the fact that the keep-alive ping mode is a powerful tool for monitoring session agents' health, you might want to use the continuous ping method if you are concerned about the Oracle Enterprise Session Border Controller not distinguishing between unsuccessful responses from next-hop session agents and ones from devices downstream from the next-hop session agent. For example, if a SIP hop beyond the session agent responds with a 503 Service Unavailable, the Oracle Enterprise Session Border Controller does not detect whether a session agent or the device beyond it generated the response.

When you use the continuous ping method, only the next-hop session agent responds—preventing the request from being sent to downstream devices. The Oracle Enterprise Session Border Controller also sends the ping in regular traffic conditions when in continuous ping mode, so it is certain the response comes from the next hop associated with the session agent. And in continuous ping mode, only entries for the ping-out-service-response-codes parameter and transaction timeouts bring session agents OOS.

SIP SA Continuous Ping Configuration

You can set the ping mode in the session agent or session constraints configuration. For backward compatibility, the default for the ping-send-mode parameter is keep-alive, or the functionality available in Release C5.1.0 and in earlier releases.

To configure the ping mode for a session agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. ping-send-mode—If to want to use continuous ping mode to send ping messages to session agents in regular traffic conditions, set this parameter to continuous. If you want to use the keep-alive mode, leave this parameter set to keep-alive (default).
5. Save and activate your configuration.

   To configure the ping mode for the session constraints:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

8. Type session-constraints and press Enter.

```
ACMEPACKET(session-router)# session-constraints
ACMEPACKET(session-constraints)#
```

   If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

9. ping-send-mode—If to want to use continuous ping mode to send ping messages to session agents in regular traffic conditions, set this parameter to continuous. If you want to use the keep-alive mode, leave this parameter set to keep-alive (default).
10. Save and activate your configuration.

H.323 Session Agents

H.323 session agents can include the following:

- Gatekeepers
- Gateways
- MCUs

Overlapping H.323 Session Agent IP Address and Port

You can now configure H.323 session agents to use overlapping IP addresses.

H.323 session agents continue are identified by their hostname when used in referencing configuration parameters—such as local policy next hops and session agent group destinations. This is why the hostname must be unique. However, when the Oracle Enterprise Session Border Controller selects a session agent to use, it chooses the appropriate realm and H.323 stack based on the hostname. This is the case even if there are other session agents with the same IP address and port. Likewise, incoming calls are matched to the session agent based on the incoming realm.

There are no specific parameters to configure in order to enable this feature. For it to work properly, however, each H.323 session agent must be configured with a unique hostname (still the primary index). Otherwise, session agents with non-unique hostnames will overwrite one another.

To create overlapping H.323 session agents, you give each of them a unique hostname, which only serves to identify each individually. The Oracle Enterprise Session Border Controller subsequently uses this label as the next hop destination in relevant local policy route entries.

Managing Session Agent Traffic

The Oracle Enterprise Session Border Controller monitors availability, session load, and session rate for each session agent in real time. The session agent's state is determined by its performance relative to the constraints applied to it and its availability.

The following table lists the conditions that cause the Oracle Enterprise Session Border Controller to suspend the routing of traffic to a session agent, along with the criteria for restoring the route.

Constraint Condition	SIP Criteria	H.323 Criteria	Action	Criteria for Resuming
Maximum sessions exceeded	Maximum concurrent SIP sessions exceeded.	Maximum concurrent H.323 sessions exceeded.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum sessions value.

Constraint Condition	SIP Criteria	H.323 Criteria	Action	Criteria for Resuming
		If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.		
Maximum outbound sessions exceeded	Maximum concurrent outbound SIP sessions exceeded.	Maximum concurrent outbound H.323 sessions exceeded. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum outbound sessions value.
Maximum burst rate exceeded	Maximum burst rate exceeded in current window.	Maximum burst rate exceeded in current window. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	Session agent is declared in constraint violation state.	Burst rate in subsequent window drops below maximum burst rate.
Maximum sustained rate exceeded	Maximum sustained rate exceeded in current window.	Maximum burst rate exceeded in current window. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum	Session agent is declared in constraint violation state.	Sustained rate in subsequent window drops below the maximum sustained rate.

Constraint Condition	SIP Criteria	H.323 Criteria	Action	Criteria for Resuming
		outbound sessions are measured.		
Session agent unavailable or unresponsive	SIP transaction expire timer expires for any out-of-dialogue request. For example, INVITE, REGISTER, or ping.	Response timer expires. The default is T301=4 seconds. Connect timer expires. The default is T303=32 seconds. If the session agent is a peer gatekeeper, the LRQ response time is used to determine availability. The RAS response timer is 4 seconds.	Session agent is declared in constraint violation state or out-of-service. The time to resume timer starts.	Time to resume timer expires and the Oracle Enterprise Session Border Controller declares the session agent in-service. or Session agent responds to subsequent pings (SIP only).

Session Agent Groups

Session agent groups contain individual session agents. Members of a session agent group are logically equivalent (although they might vary in their individual constraints) and can be used interchangeably. You can apply allocation strategies to session agent groups.

Examples of session agent groups include the following:

- application server cluster
- media gateway cluster
- softswitch redundant pair
- SIP proxy redundant pair
- gatekeeper redundant pair

Session agent group members do not need to reside in the same domain, network, or realm. The Oracle Enterprise Session Border Controller can allocate traffic among member session agents regardless of their location. It uses the allocation strategies configured for a SAG to allocate traffic across the group members.

Allocation strategies include the following:

Allocation Strategy	Description
Hunt	Oracle Enterprise Session Border Controller selects the session agents in the order in which they are configured in the SAG. If the first agent is available, and has not exceeded any defined constraints, all traffic is sent to the first agent. If the first agent is unavailable, or is in violation of constraints, all traffic is sent to the second agent. And so on for all session agents in the SAG. When the first agent returns to service, the traffic is routed back to it.
Round robin	Oracle Enterprise Session Border Controller selects each session agent in the order in which it is configured, routing a session to each session agent in turn.

Allocation Strategy	Description
Least busy	Oracle Enterprise Session Border Controller selects the session agent with the least number of active sessions, relative to the maximum outbound sessions or maximum sessions constraints (lowest percent busy) of the session agent.
Proportional distribution	Session agents are loaded proportionately based upon the respective maximum session constraint value configured for each session agent.
Lowest sustained rate	Oracle Enterprise Session Border Controller routes traffic to the session agent with the lowest sustained session rate, based on observed sustained session rate.

You apply allocation strategies to select which of the session agents that belong to the group should be used. For example, if you apply the Hunt strategy session agents are selected in the order in which they are listed.

Request URI Construction as Sent to SAG-member Session Agent

The Oracle Enterprise Session Border Controller constructs the request URI for a session agent selected from a session agent group by using the **session-agent** > **hostname** value of the selected **session-agent** target. This default behavior enables features such as trunk groups and ENUM to work properly. However, care must be given when the **hostname** parameter is not a resolvable FQDN. The **sag-target-uri**=<value> option can be used to overcome the default behavior.

The value is either

- **ip** – request URI constructed from **session-agent** > **ip-address**
- **host** – request URI constructed from **session-agent** > **hostname**

This option is global and is configured in the **sip-config** configuration element.

Request URI Construction as Forwarded to SAG-member Session Agent

The Oracle Enterprise Session Border Controller constructs the request URI for a session agent selected from a session agent group by using the **session-agent** > **hostname** value of the selected **session-agent** target. This default behavior enables features such as trunk groups and ENUM to work properly. However, care must be given when the **hostname** parameter is not a resolvable FQDN. Use the **sag-target-uri=**<value> option to override the default behavior.

The value can be set to either:

- **ip** - request URI constructed from **session-agent** > **ip-address**
- **host** - request URI constructed from **session-agent** > **hostname**

This option is global and is configured in the **sip-config** configuration element.

SIP Session Agent Group Recursion

You can configure a SIP session agent group (SAG) to try all of its session agents rather than to the next-best local policy match if the first session agent in the SAG fails.

With this feature disabled, the Oracle Enterprise Session Border Controller performs routing by using local policies, trunk group URIs, cached services routes, and local route tables. Local policies and trunk group URIs can use SAGs to find the most appropriate next-hop session agent based on the load balancing scheme you choose for that SAG: round robin, hunt, proportional distribution, least busy, and lowest sustained rate. When it locates a SAG and selects a specific session agent, the Oracle Enterprise Session Border Controller tries only that single session agent. Instead of trying other members of the SAG, the Oracle Enterprise Session Border Controller recurses to the local policy that is the next best match. This happens because the Oracle Enterprise Session Border Controller typically chooses a SAG based on the fact that it has not breached its constraints, but the Oracle Enterprise Session Border Controller only detects failed call attempts (due to unreachable next hops, unresolved ENUM queries, or SIP 4xx/5xx/6xx failure

responses) after it has checked constraints. So the Oracle Enterprise Session Border Controller only re-routes if there are additional matching local policies.

When you enable SIP SAG recursion, the Oracle Enterprise Session Border Controller will try the additional session agents in the selected SAG if the previous session agent fails. You can also set specific response codes in the SAG configuration that terminate the recursion. This method of terminating recursion is similar to the Oracle Enterprise Session Border Controller's ability to stop recursion for SIP interfaces and session agents.

Session agents are selected according to the strategy you set for the SAG, and these affect the way that the Oracle Enterprise Session Border Controller selects session agents when this feature enabled:

- Round robin and hunt—The Oracle Enterprise Session Border Controller selects the first session agent according to the strategy, and it selects subsequent session agents based on the order they are entered into the configuration.
- Proportional distribution, least busy, and lowest sustained rate—The Oracle Enterprise Session Border Controller selects session agents based on the list of session agents sorted by the criteria specified.

You can terminate recursion based on SIP response codes that you enter into the SAG configuration. You can configure a SAG with any SIP response code in the 3xx, 4xx, and 5xx groups. Since you can also set such a list in the session agent configuration, this list is additive to that one so that you can define additional codes for a session agent group with out having to repeat the ones set for a session agent.

About Local Policy

This section explains the role of local policy. Local policy lets you indicate where session requests, such as SIP INVITES, should be routed and/or forwarded. You use a local policy to set a preference for selecting one route over another. The local policy contains the following information that affects the routing of the SIP and H.323 signaling messages:

- information in the From header

  Information in the message's From header is matched against the entries in the local policy's from address parameter to determine if the local policy applies.
- list of configured realms

  This list identifies from what realm traffic is coming and is used for routing by ingress realm. The source realms identified in the list must correspond to the valid realm IDs you have already configured
- local policy attributes

  The attributes serve as an expression of preference, a means of selecting one route over another. They contain information such as the next signaling address to use (next hop) or whether you want to select the next hop by codec, the realm of the next hop, and the application protocol to use when sending a message to the next hop. You can also use the attributes to filter specific types of traffic.

Routing Calls by Matching Digits

Local policy routing of a call can be based on matching a sequence of digits against what is defined in the local policy. This sequence refers to the first digits in the (phone) number, matching left to right.

The following examples show how the Oracle Enterprise Session Border Controller matches an area code or number code against configured local policies.

- If the number or area code being matched is 1234567 (where 123 is an area code), and the from address value in one local policy is 123, and the from address value in another local policy is 12, the Oracle Enterprise Session Border Controller forwards the call to the server that is defined as the next hop in the local policy with 123 as the from address value.
- If the number or area code being matched is 21234, and the from address value in one local policy is 123, and the from address value in another local policy is 12, the Oracle Enterprise Session Border Controller will not find a match to either local policy because the first character of the number or area code must match the first character in a from address or to address field.

The following examples show how the Oracle Enterprise Session Border Controller matches an area or number code against different local policies: the first one has a From address value of 12 and the second has a From address value of 123. The Oracle Enterprise Session Border Controller chooses the route of the local policy that is configured with the most digits matching the area or number code in its From address and To address fields.

- When the two different local policies route to two different servers, and the area or number code being matched is 123, the Oracle Enterprise Session Border Controller selects the second local policy based on the From address value of 123.
- When the two different local policies route to two different servers, and the area or number code being matched is 124, the Oracle Enterprise Session Border Controller selects the first local policy based on the From address value of 12.

SIP and H.323 Interworking

You need to configure local policies, including the requisite local policy attributes, to use the H.323<—>SIP interworking (IWF). Flow progression in H.323<—>SIP traffic depends heavily on the local policies configured for the Oracle Enterprise Session Border Controller, which determine what protocol is used on the egress side of a session.

You set the application protocol (an local policy attribute option) to instruct the Oracle Enterprise Session Border Controller to interwork the protocol of an ingress message into a different protocol (H.323<—>SIP or SIP—>H.323) upon its egress to the next hop.

For example, if the application protocol is set to SIP, an inbound H.323 message will be interworked to SIP as it is sent to the next hop. An inbound SIP message would pass to the next hop unaffected. If the application protocol is set to H323, an inbound SIP message will be interworked to H.323 before being sent to the next hop.

Route Preference

The Oracle Enterprise Session Border Controller builds a list of possible routes based on the source realm and the From-address (From-URI) and To-address (Request-URI), which forms a subset from which preference then decides. Any local policy routes currently outside of the configured time/day are not used, if time/day are set. Also, any local policy routes not on the list of carriers (if carriers is set and the requests has a Carrier header) are not used.

Note: Source realm is used in the local policy lookup process, but it is not used in route preference calculations.

The Oracle Enterprise Session Border Controller applies preference to configured local policies in the following order:

1. Cost (cost in local policy attributes) is always given preference.
2. Matching media codec (media profiles option in local policy attributes).
3. Longest matching To address (to address list in local policy).
4. Shortest matching To address (to address list in local policy).
5. Longest matching From address (from address list in local policy).
6. Shortest matching From address (from address list in local policy).
7. Narrowest/strictest day of week specification (days of week option in local policy attributes).
8. Narrowest/strictest time of day specification (start time and end time options in local policy attributes).
9. Wildcard matches (use of an asterisk as a wildcard value for the from address and to address lists in local policy).
10. Wild card matches are given the least preference. A prefix value of 6 is given a higher preference than a prefix value of * even though both prefix values are, in theory, the same length.

DTMF-Style URI Routing

The Oracle Enterprise Session Border Controller supports the alphanumeric characters a-d, A-D, the asterisk (*), and the ampersand (#) for local policy matching purposes. The Oracle Enterprise Session Border Controller handles these characters as standards DN (POTS) or FQDN when found in the to-addr (req-uri username) or from-addr (from0uri username for SIP, SIPS, and TEL URIs.

In addition, before performing the lookup match, the Oracle Enterprise Session Border Controller strips characters that provide ease-of-reading separation. For example, if the Oracle Enterprise Session Border Controller were to receive a req-uri containing tel:a-#1-781-328-5555, it would treat it as tel:a#17813285555.

SIP Routing

This section describes SIP session routing. When routing SIP call requests, the Oracle Enterprise Session Border Controller communicates with other SIP entities, such as SIP user devices, other SIP proxies, and so on, to decide what SIP-based network resource each session should visit next. The Oracle Enterprise Session Border Controller processes SIP call requests and forwards the requests to the destination endpoints to establish, maintain, and terminate real-time multimedia sessions.

Certain items in the messages are matched with the content of the local policy, within constraints set by the previous hop session agent, and the SIP configuration information (for example, carrier preferences) to determine a set of applicable next hop destinations.

The sending session agent is validated as either a configured session agent or a valid entry in a user cache. If the session INVITATION does not match any registering user, the SIP proxy determines the destination for routing the session INVITATION.

Limiting Route Selection Options for SIP

You can configure the local policy to use the single most-preferred route. And you can configure the SIP configuration max routes option to restrict the number of routes which can be selected from a local policy lookup:

- A max-routes=1 value limits the Oracle Enterprise Session Border Controller to only trying the first route from the list of available preferred routes.
- A max-routes=0 value or no max-routes value configured in the options field allows the Oracle Enterprise Session Border Controller to use all of the routes available to it.

A Oracle Enterprise Session Border Controller configured for H.323 architectures will have access to all of the routes it looks up by default.

About Loose Routing

According to RFC 3261, a proxy is loose routing if it follows the procedures defined in the specification for processing of the Route header field. These procedures separate the destination of the request (present in the Request-URI) from the set of proxies that need to be visited along the way (present in the Route header field).

When the SIP NAT's route home proxy field is set to enabled, the Oracle Enterprise Session Border Controller looks for a session agent that matches the home proxy address and checks the loose routing field value. If the loose routing is:

- enabled—A Route header is included in the outgoing request in accordance with RFC 3261.
- disabled—A Route header is not included in the outgoing request; in accordance with the route processing rules described in RFC 2543 (referred to as strict routing). That rule caused proxies to destroy the contents of the Request-URI when a Route header field was present.

Whether loose routing field is enabled is also checked when a local policy 's next hop value matches a session agent. Matching occurs if the hostname or the session agent's IP address field value corresponds to the next hop value. If loose routing is enabled for the matching session agent, the outgoing request retains the original Request-URI and Route header with the next hop address.

About the Ingress Realm

You can create a list of realms in your local policy that is used by the Oracle Enterprise Session Border Controller to determine how to route traffic. This list determines from which realm traffic is coming and is used for routing by ingress realm.

The source realm values must correspond to valid identifier entered when the realm was configured.

About the Egress Realm

An egress realm allows SIP signaling to travel out of the Oracle Enterprise Session Border Controller through a network other than the home realm. The Oracle Enterprise Session Border Controller uses egress realms for signaling purposes (when matching flows). When a packet arrives at the Oracle Enterprise Session Border Controller with a destination address that does not match any defined session agents, the Oracle Enterprise Session Border Controller uses the address associated with the realm that is, in turn, associated with the SIP configuration's egress realm ID, as the outgoing network. With the use of the egress realm ID, it is possible to define a default route for SIP requests addressed to destinations outside the home realm. If no egress realm is defined, the home realm (default ingress realm) is used as the default egress realm.

With session agent egress realm configured, the Oracle Enterprise Session Border Controller adds a default egress realm to the session agent to identify the signaling interface used for ping requests. The Oracle Enterprise Session Border Controller also uses the default egress realm when the normal routing request does not yield an egress realm—for example, when a local policy does not specify the next hop's realm.

When you configure session agents, you can define them without realms or you can wildcard the realm value. These are global session agents, and multiple signaling interfaces can reach them. Then, when you use session agent pinging, the Oracle Enterprise Session Border Controller sends out ping requests using the signaling interface of the default egress realm defined in the global SIP configuration. The global session agents in certain environments can cause problems when multiple global session agents residing in multiple networks, some of which might not be reachable using the default SIP interface egress realm.

The Oracle Enterprise Session Border Controller uses the session agent egress realm for ping messages even when the session agent has a realm defined. For normal request routing, the Oracle Enterprise Session Border Controller uses the egress realm for global session agents when local policies or SIP-NAT bridge configurations do not point to an egress realm.

Ping Message Egress Realm Precedence

For ping messages, the egress realm precedence occurs in the following way (in order of precedence):

- Egress realm identified for the session agent.
- Session agent realm (set in the realm-id parameter) or the wildcarded value
- Global SIP configuration egress realm, when configured in the egress-realm parameter
- Global SIP configuration home realm

Normal Request Egress Realm Precedence

For normal request routing, the egress realm precedence occurs in the following way (in order of precedence):

- Egress SIP-NAT realm, when the route-home-proxy parameter is set to forced and no local policy match is found
- Matching local policy realm, when configured in the local policy attributes
- Session agent realm (set in the realm-id parameter) or the wildcarded value
- Session agent egress realm, when configured in the egress-realm-id parameter
- Global SIP configuration egress realm, when configured in the egress-realm parameter
- Global SIP configuration home realm

Session Agent Egress Realm Configuration

Configuring a session agent egress realm is optional.

To configure a session agent egress realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

   If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI select command) before making your changes.

4. egress-realm-id—Enter the name of the realm you want defined as the default egress realm used for ping messages. The Oracle Enterprise Session Border Controller will also use this realm when it cannot determine the egress realm from normal routing. There is no default value for this parameter.
5. Save and activate your configuration.

About SIP Redirect

SIP redirect involves proxy redirect and tunnel redirect.

Proxy Redirect

You can configure the SIP proxy mode to define how the SIP proxy will forward requests coming from the session agent. This value is used if the session agent's proxy mode has no value (is empty).

Tunnel Redirect

You can use tunnel redirect when requests are routed to a server behind a SIP NAT that sends redirect responses with addresses that should not be modified by the SIP NAT function. For example, a provider might wish to redirect certain calls (like 911) to a gateway that is local to a the UA that sent the request. Since the gateway address is local to the realm of the UA, it should not be modified by the SIP NAT of the server's realm. Note that the server must have a session agent configured with the redirect-action field set to the proxy option in order to cause the redirect response to be sent back to the UA.

SIP Method Matching and To Header Use for Local Policies

For SIP, this feature grants you greater flexibility when using local policies and has two aspects: basing local policy routing decisions on one or more SIP methods you configure and enabling the Oracle Enterprise Session Border Controller to use the TO header in REGISTER messages for routing REGISTER requests.

SIP Methods for Local Policies

This feature allows the Oracle Enterprise Session Border Controller to include SIP methods in routing decisions. If you want to use this feature, you set a list of one or more SIP methods in the local policy attributes. These are the SIP methods you can enter in the list: INVITE, REGISTER, PRACK, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, UPDATE, MESSAGE, and PUBLISH.

After the Oracle Enterprise Session Border Controller performs a local policy look-up for SIP, it then searches for local policy attributes that have this methods list configured. If it finds a a set of policy attributes that matches a method that matches the traffic it is routing, the Oracle Enterprise Session Border Controller uses that set of policy attributes. This means that the Oracle Enterprise Session Border Controller considers first any policy attributes with methods configured before it considers those that do not have methods. In the absence of any policy attributes with methods, the Oracle Enterprise Session Border Controller uses the remaining ones for matching.

In cases where it finds neither matching policy attributes with methods or matching policy attributes without them, the Oracle Enterprise Session Border Controller either rejects the calls with a 404 No Routes Found (if the request calls for a response) or drops the call.

You configure local policy matching with SIP methods in the local policy attributes parameter calls methods. This parameter is a list that takes either one or multiple values. If you want to enter multiple values, you put them in the same command line entry, enclosed in quotation marks and separated by spaces.

To configure SIP methods for local policy matching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type local-policy and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type policy-attributes and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy))# policy-attributes
ACMEPACKET(policy-attributes)#
```

5. methods—Enter the SIP methods you want to use for matching this set of policy attributes. Your list can include: INVITE, REGISTER, PRACK, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, UPDATE, MESSAGE, and PUBLISH.

   By default, this parameter is empty—meaning that SIP methods will not be taken into consideration for routing based on this set of policy attributes.

   If you want to enter more than one method, you entry will resemble the following example.

```
ACMEPACKET(local-policy-attributes)# methods "PRACK INFO REFER"
```

6. Save and activate your configuration.

Routing Using the TO Header

For the Oracle Enterprise Session Border Controller's global SIP configuration, you can enable the use of an ENUM query to return the SIP URI of the Registrar for a SIP REGISTER message. Without this feature enabled, the Oracle Enterprise Session Border Controller uses the REQUEST URI. This ability can be helpful because REGISTER messages only have the domain in the REQUEST URI, whereas the SIP URI in the To header contains the user's identity.

There are two parts to enabling this feature. First, you must enable the register-use-to-for-lp parameter in the global SIP configuration. Then you can set the next-hop in the applicable local policy attributes set to ENUM.

To enable your global SIP configuration for routing using the TO header:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. If you are adding this feature to a pre-existing SIP configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. register-use-to-for-lp—Set this parameter to enabled if you want the Oracle Enterprise Session Border Controller to use, for routing purposes, an ENUM query to return the SIP URI of the Registrar for a SIP REGISTER message. This parameters defaults to disabled.

   To set up your local policy attributes for routing using the TO header:

5. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

6. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

7. Type local-policy and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

8. Type policy-attributes and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy))# policy-attributes
ACMEPACKET(policy-attributes)#
```

9. next-hop—This is the next signaling host. Set this parameter to ENUM if you want to use SIP methods in local policy attribute information for routing purposes.
10. Save and activate your configuration.

H.323 Routing

This section describes H.323 routing.

Egress Stack Selection

Egress stack selection includes static stack selection and policy-based stack selection

Static Stack Selection

In static stack selection, the outgoing stack is determined though the establishment of associated stacks in the h323 stack.

The incoming stack (configured in the h323 stack) uses its associated stack value to determine the associated outgoing stack. The associated stack value corresponds to the name of an h323 stack. This type of selection is referred to as static because the incoming stack always uses the stack specified in the associated stack as the outgoing stack; no other stacks are considered.

Policy-Based Stack Selection

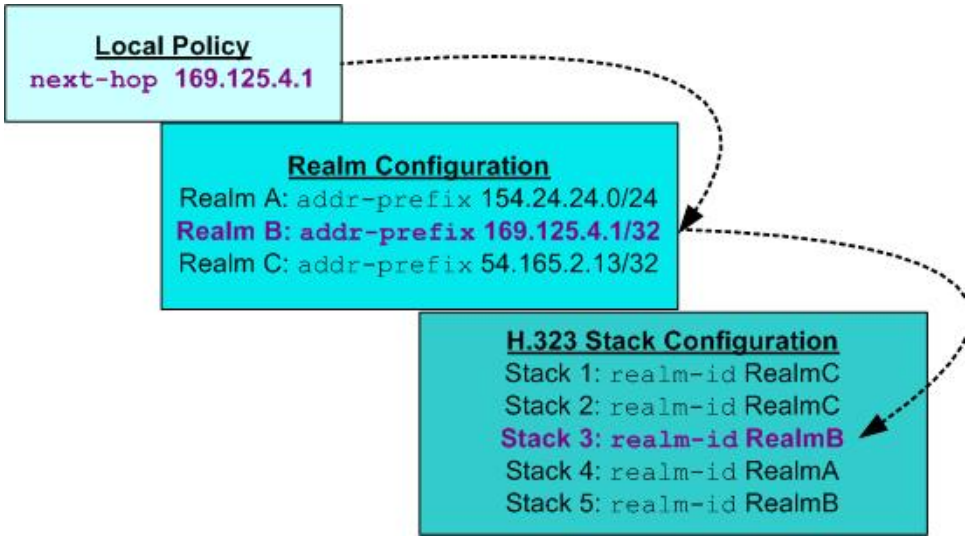
The Oracle Enterprise Session Border Controller performs dynamic, policy-based stack selection when an H.323 call arrives at the Oracle Enterprise Session Border Controller and a configured associated outgoing stack cannot be found.

For policy-based stack selection, the Oracle Enterprise Session Border Controller refers to local policies that contain address information that corresponds to incoming traffic. This information is contained in the local policy's To address and From address fields. For the source, this information is matched with the Q.931 calling party number; if there is no calling party number, the H.323 source address is used. For the destination, this information is matched with the called party number; if there is no called party number, then the H.323 destination address is used.

After a local policy corresponding to the incoming traffic has been found, the Oracle Enterprise Session Border Controller looks at the next hop value (a local policy attribute) and selects a local policy for the basis of stack selection. If the local policy look-up yields multiple local policies with the same next hop values, but with different cost values, the local policy with the lowest cost value is selected.

If a realm is not defined in the local policy, the next hop address is then matched against the address prefix values for the realms that are configured for the system. Thus, the Oracle Enterprise Session Border Controller discovers the realm for this traffic. Using this realm information, the Oracle Enterprise Session Border Controller performs stack selection. It uses the first configured H.323 stack in the Oracle Enterprise Session Border Controller's configuration that has a realm ID value matching the identifier field of the realm with the appropriate address prefix.

In the following example, the local policy matching yields a local policy with a next hop value of 169.125.4.1, which corresponds to RealmB. The outgoing stack selected is Stack 3 because it is the first stack to have been configured with RealmB as the realm ID.



Policy-Based Stack Selection

Registration Caching

The Oracle Enterprise Session Border Controller can cache and proxy an H.225 RegistrationRequest (RRQ) between an H.323 endpoint and a gatekeeper. Registration caching serves two functions:

- It allows aggregation of RRQs sent to a gatekeeper stack and proxies those requests through the gateway stack. If the external gatekeeper associated with the gatekeeper stack supports additive registration, the requests will be consolidated. Furthermore, if the gatekeeper supports additive registration, the Oracle Enterprise Session Border Controller will register in an additive manner, meaning that will send additive RRQs.
- It allows the gatekeeper stack to use the registration information to route calls from other realms to endpoints in its realms.

To perform registration caching, the Oracle Enterprise Session Border Controller must be configured with at least two stacks. One of these stacks will receive registrations (gatekeeper stack), and one stack will proxy registrations (gateway stack). The Oracle Enterprise Session Border Controller caches all successful registrations and uses the cache to route calls to the endpoints.

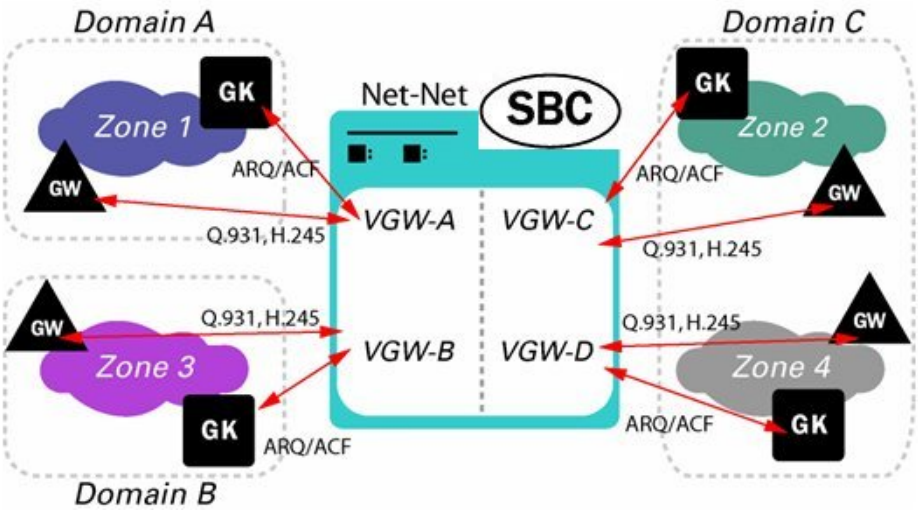
Gatekeeper Provided Routes

Gatekeeper provided routes includes back-to-back gateways, back-to-back gatekeeper and gateway, and interworking gatekeeper/gateway.

Back-to-Back Gateway

When the Oracle Enterprise Session Border Controller is functioning as a back-to-back gateway (B2BGW), it appears as multiple H.323 gateways to multiple networks. Each Oracle Enterprise Session Border Controller virtual gateway discovers and registers with a gatekeeper in its respective domain. Each gateway relies on its gatekeeper for admission and location services through the ARQ/ACF exchange. H.225 call control and H.245 messages are exchanged directly with the terminating gateway or gatekeeper. Routing policies are used to associate one virtual gateway with another.

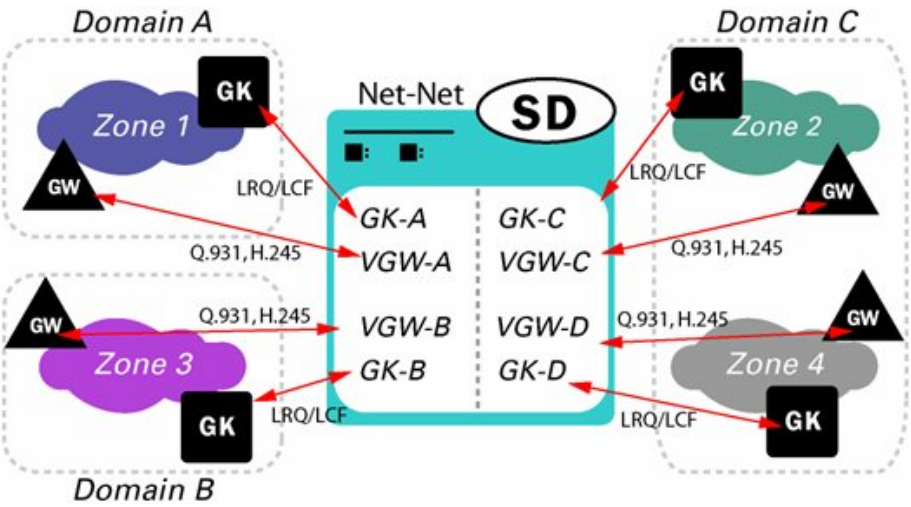
The following diagram illustrates the back-to-back gateway.



Back-to-Back Gatekeeper and Gateway

For peering connections where both networks use inter-domain gatekeeper signaling, the Oracle Enterprise Session Border Controller is configured as a back-to-back gatekeeper proxy and gateway mode of operation. The Oracle Enterprise Session Border Controller responds and issues LRQs and LCFs/LRJs acting as a routed gatekeeper. Peered gatekeepers send LRQ to the RAS address of one of the Oracle Enterprise Session Border Controller's virtual gatekeepers and it responds by providing its call signaling address that performs the gateway functions. Routing policies are used to determine the egress virtual gatekeeper that then exchanges LRG/LCF to determine the call signaling address of the terminating gateway.

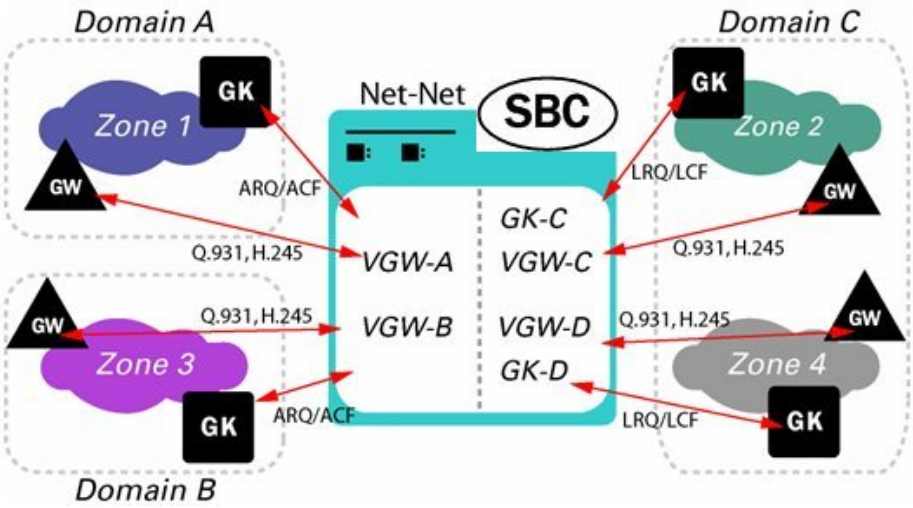
The following diagram illustrates the back-to-back gatekeeper and gateway.



Interworking Gatekeeper Gateway

In the interworking gatekeeper/gateway signaling mode of operation, the Oracle Enterprise Session Border Controller interworks between the other two modes; presenting a routed gatekeeper interface to one zone and a gateway to the other.

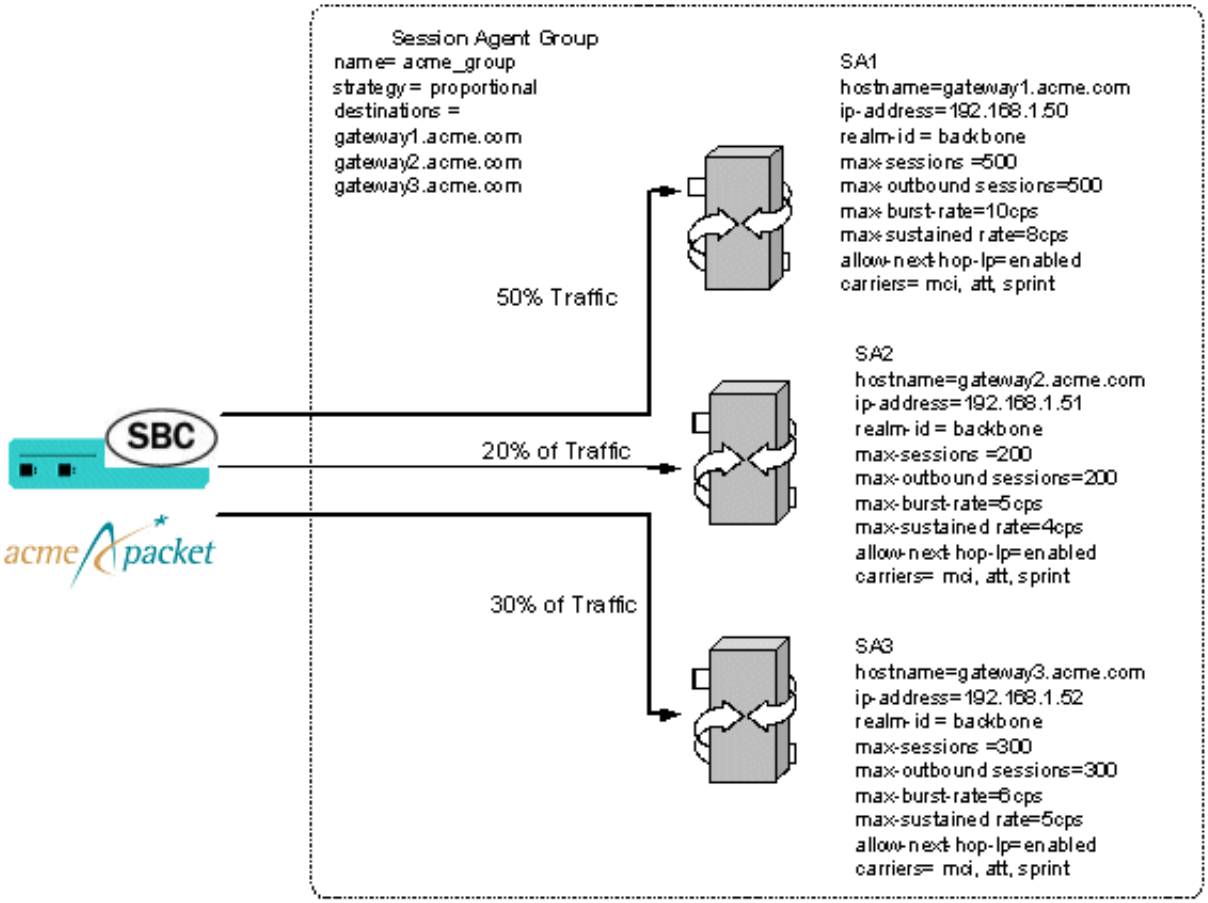
The following diagram illustrates the interworking gatekeeper/gateway.



Load Balancing

This section describes Oracle Enterprise Session Border Controller load balancing. You can use session agent groups to assist in load balancing among session agents. You define concurrent session capacity and rate attributes for each session agent and then define the session agent group. Next, you select the allocation strategy you want applied to achieve the load balancing you want.

The following example shows a configuration for load balancing gateways based on a proportional allocation strategy.



Routing and load balancing capabilities include the following:

- least cost, which includes cost-based and time-based routing
- customer preference
- traffic aggregation
- routing by media (codec) type
- capacity-based, by destination
- service element load balancing
- service element failure detection and re-route
- session agent failure
- routing by codec

Configuring Routing

This section explains how to configure routing on the Oracle Enterprise Session Border Controller.

Configuration Prerequisite

You should have already configured the realms for your environment before you configure the routing elements. You need to know the realm identifier when configuring session agents and local policy.

You can use an asterisk (*) when the session agent exists in multiple realms.

Configuration Order

Recommended order of configuration:

- realm
- session agent
- session agent group
- local policy

Routing Configuration

You can enable, then configure, individual constraints that are applied to the sessions sent to the session agent. These constraints can be used to regulate session activity with the session agent. In general, session control constraints are used for session agent groups or SIP proxies outside or at the edge of a network. Some individual constraints, such as maximum sessions and maximum outbound sessions are not applicable to core proxies because they are transaction stateful, instead of session stateful. Other constraints, such as maximum burst rate, burst rate window, maximum sustained rate, and sustained rate are applicable to core routing proxies.

Configuring Session Agents

To configure session agents:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. host-name—Enter the name of the host associated with the session agent in either hostname or FQDN format, or as an IP address.

If you enter the host name as an IP address, you do not have to enter an IP address in the optional IP address parameter. If you enter the host name in FQDN format, and you want to specify an IP address, enter it in the optional IP address parameter. Otherwise you can leave the IP address parameter blank to allow a DNS query to resolve the host name.

If the initial DNS query for the session agent fails to get back any addresses, the session agent is put out-of-service. When session agent is pinged, the DNS query is repeated. The ping message is not sent until the DNS query gets back one or more IP addresses. After the query receives some addresses, the ping message is sent. The session agent remains out of service until one of the addresses responds.

Note: The value you enter here must be unique to this session agent. No two session agents can have the same hostname.

The hostnames established in the session agent populate the corresponding fields in other elements.

5. ip-address—Optional. Enter the IP address for the hostname you entered in FQDN format if you want to specify the IP address. Otherwise, you can leave this parameter blank to allow a DNS query to resolve the host name.
6. port—Enter the number of the port associated with this session agent. Available values include:
   - zero (0)—If you enter zero (0), the Oracle Enterprise Session Border Controller will not initiate communication with this session agent (although it will accept calls).
   - 1025 through 65535

   The default value is 5060.

   Note: If the transport method value is TCP, the Oracle Enterprise Session Border Controller will initiate communication on that port of the session agent.
7. state—Enable or disable the session agent by configuring the state. By default, the session agent is enabled.
   - enabled | disabled
8. app-protocol—Enter the protocol on which you want to send the message. The default value is SIP. Available values are:
   - SIP | H.323
9. app-type—If configuring H.323, indicate whether the application type is a gateway or a gatekeeper. Available values include:
   - H.323-GW—gateway
   - H.323-GK—gatekeeper
10. transport-method—Indicate the IP protocol to use (transport method) to communicate with the session agent. UDP is the default value. The following protocols are supported:
    - UDP—Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
    - UDP+TCP—Allows an initial transport method of UDP, followed by a subsequent transport method of TCP if and when a failure or timeout occurs in response to a UDP INVITE. If this transport method is selected, INVITEs are always sent through UDP as long as a response is received.
    - DynamicTCP—dTCP indicates that dynamic TCP connections are the transport method for this session agent. A new connection must be established for each session originating from the session agent. This connection is torn down at the end of a session.
    - StaticTCP—sTCP indicates that static TCP connections are the transport method for this session agent. Once a connection is established, it remains and is not torn down.
    - DynamicTLS—dTLS indicates that Dynamic TLS connections are the transport method for this session agent. A new connection must be established for each session originating from the session agent. This connection is torn down at the end of a session.
    - StaticTLS—sTLS indicates that Static TLS connections are the transport method for this session agent. Once a connection is established, it will remain and not be torn down.
11. realm-id—Optional. Indicate the ID of the realm in which the session agent resides.

The realm ID identifies the realm for sessions coming from or going to this session agent. For requests coming from this session agent, the realm ID identifies the ingress realm. For requests being sent to this session agent, the realm ID identifies the egress realm. In a Oracle Enterprise Session Border Controller, when the ingress and egress realms are different, the media flows must be steered between the realms.

- no value: the egress realm is used unless the local policy dictates otherwise
- asterisk (*): keep the egress realm based on the Request URI

Note: The realm ID you enter here must match the valid identifier value entered when you configured the realm.

12. description—Optional. Enter a descriptive name for this session agent.
13. carriers—Optional. Add the carriers list to restrict the set of carriers used for sessions originating from this session agent.

    Carrier names are arbitrary names that can represent specific service providers or traditional PSTN telephone service providers (for sessions delivered to gateways). They are global in scope, especially if they are exchanged in TRIP. Therefore, the definition of these carriers is beyond the scope of this documentation.

    You could create a list using carrier codes already defined in the North American Numbering Plan (NANP); or those defined by the local telephone number or carrier naming authority in another country.

    Note: If this list is empty, any carrier is allowed. If it is not empty, only local policies that reference one or more of the carriers in this list will be applied to requests coming from this session agent.
14. allow-next-hop-lp—Indicate whether this session agent can be used as a next hop in the local policy.

    If you retain the default value of enabled, the session agent can be used as the next hop for the local policy. Valid values are:

    - enabled | disabled
15. constraints—Enable this parameter to indicate that the individual constraints you configure in the next step are applied to the sessions sent to the session agent. Retain the default value of disabled if you do not want to apply the individual constraints. Valid values are:

    - enabled | disabled

    Note: In general, session control constraints are used for SAGs or SIP proxies outside or at the edge of a network.
16. Enter values for the individual constraints you want applied to the sessions sent to this session agent. The following table lists the available constraints along with a brief description and available values.

Constraint	Description
maximum sessions	Maximum number of sessions (inbound and outbound) allowed by the session agent. The range of values is: • minimum: zero (0) is the default value and means there is no limit • maximum: 4294967295
maximum outbound sessions	Maximum number of simultaneous outbound sessions (outbound from the Oracle Enterprise Session Border Controller) that are allowed from the session agent. The range of values is: • minimum: zero (0) is the default value and means there is no limit • maximum: 4294967295 The value you enter here cannot be larger than the maximum sessions value.
maximum burst rate	Number of session invitations allowed to be sent to or received from the session agent within the configured burst rate window value. SIP session invitations arrive at and leave from the session agent in

Constraint	Description
	intermittent bursts. By entering a value in this field, you can limit the amount of session invitations that are allowed to arrive at and leave from the session-agent. For example, if you enter a value of 50 here and a value of 60 (seconds) for the burst rate window constraint, no more than 50 session invitations can arrive at or leave from the session agent in that 60 second time frame (window). Within that 60-second window, any sessions over the limit of 50 are rejected. The range of values is: • minimum: zero (0) session invitations per second • maximum: 4294967295 session invitations per second Zero is the is the default value.
maximum sustain rate	Maximum rate of session invitations (per second) allowed to be sent to or received from the session agent within the current window. The current rate is determined by counting the number of session invitations processed within a configured time period and dividing that number by the time period. By entering a value in this field, you can limit the amount of session invitations that are allowed to arrive at and leave from the session agent over a sustained period of time. For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes. For example, if you enter a value of 5000 here and a value of 3600 (seconds) for the sustain rate window constraint, no more than 5000 session invitations can arrive at or leave from the session agent in any given 3600 second time frame (window). Within that 3600-second window, sessions over the 5000 limit are rejected. The range of values is: • minimum: zero (0) invitations per second • maximum: 4294967295 invitations per second Zero is the is the default value. The value you set here must be larger than the value you set for the maximum burst rate constraint.
time to resume	Time in seconds after which the SIP proxy resumes sending session invitations to this session agent. This value only takes effect when the SIP proxy stops sending invitations because a constraint is exceeded. The range of values is: • minimum: zero (0) seconds • maximum: 4294967295 seconds Default is zero.
time to resume (ttr) no response	Delay in seconds that the SIP proxy must wait from the time that it sends an invitation to the session agent and gets no response before it tries again.

Constraint	Description
	The range of values is: • minimum: zero (0) seconds • maximum: 4294967295 seconds Default is zero. The value you enter here must be larger than the value you enter for the time to resume constraint.
in service period	Amount of time in seconds the session agent must be operational (once communication is re-established) before the session agent is declared as being in-service (ready to accept session invitations). This value gives the session agent adequate time to initialize. The range of values is: • minimum: zero (0) seconds • maximum: 4294967295 seconds Default is zero.
burst rate window	Burst window period (in seconds) that is used to measure the burst rate. The term window refers to the period of time over which the burst rate is computed. Refer to the maximum burst rate information. The range of values is: • minimum: zero (0) seconds • maximum: 4294967295seconds Zero is the is the default value. The value you set here must be smaller than the value you set for the maximum burst rate constraint.
sustain rate window	Sustained window period (in seconds) that is used to measure the sustained rate. Refer to the maximum sustain rate information. The range of values is: • minimum: zero (0) seconds • maximum: 4294967295seconds Zero is the is the default value. The value you set here must be larger than the value you set for the maximum sustain rate constraint.

17. req-uri-carrier-mode—SIP only. Set whether you want the selected carrier (determined by a value in the local policy) added to the outgoing message by configuring the request uri carrier mode parameter.

    You can set this parameter to let the system perform simple digit translation on calls sent to gateways. A 3-digit prefix is inserted in front of the telephone number (the Request-URI) that the gateway will use to select a trunk group. Most often, the Oracle Enterprise Session Border Controller needs to insert the carrier code into the signaling message that it sends on.

    The default value is none. The following lists the available modes.

    - none—Carrier information will not be added to the outgoing message.
    - uri-param—Adds a parameter to the Request-URI. For example, cic-XXX.

- prefix—Adds the carrier code as a prefix to the telephone number in the Request-URI (in the same manner as PSTN).

18. proxy-mode—SIP only. Indicate the proxy mode to use when a SIP request arrives from this session agent.

If this field is empty (upon initial runtime or upgrade), it's value is set to the value of the SIP configuration's proxy mode by default. If no proxy mode value was entered for the SIP configuration, the default for this field is proxy.

The following are valid proxy modes:

- proxy—If the Oracle Enterprise Session Border Controller is a Session Router, the system will proxy the request coming from the session agent and maintain the session and dialog state. If the Oracle Enterprise Session Border Controller is a Session Director, the system behaves as a B2BUA when forwarding the request.
- redirect—The system sends a SIP 3xx reDIRECT response with contacts (found in the local policy) to the previous hop.

19. redirect-action—SIP only. Indicate the action you want the SIP proxy to take when it receives a Redirect (3XX) response from the session agent.

If the response comes from a session agent and this field is empty (upon initial runtime or upgrade), the redirect action will be recurse. If no session agent is found (for example, if a message comes from an anonymous user agent), the redirect action is set to proxy. If the Redirect (3xx) response does not have any Contact header, the response will be sent back to the previous hop.

The following table lists the available proxy actions along with a brief description

- proxy—The SIP proxy passes the response back to the previous hop; based on the pfoxy mode of the original request.
- recurse—The SIP proxy serially sends the original request to the list of contacts in the Contact header of the response (in the order in which the contacts are listed in the response). For example, if the first one fails, the request will be send to the second, and so on until the request succeeds or the last contact in the Contact header has been tried.

20. loose-routing—SIP only. Enable this parameter if you want to use loose routing (as opposed to strict routing). The default is enabled. Valid values are:

- enabled | disabled

  When the SIP NAT route home proxy parameter is enabled, the Oracle Enterprise Session Border Controller looks for a session agent that matches the home proxy address and checks the loose routing value. If loose routing is enabled, a Route header is included in the outgoing request in accordance with RFC 3261. If loose routing is disabled, the Route header is not included in the outgoing request (in accordance with strict routing procedures defined in RFC 2543).

  The loose routing value is also checked when the local policy's next hop value matches a session agent. If loose routing is set to enabled, the outgoing request retains the original Request-URI and Route header with the next hop address.

21. send-media-session—SIP only. Enable this parameter if you want to include a media session description (for example, SDP) in the INVITE or REINVITE message sent by the Oracle Enterprise Session Border Controller. Setting this field to disabled prevents the Oracle Enterprise Session Border Controller from establishing flows for that INVITE message.

The default is enabled. Valid values are:

- enabled | disabled

Note: Only set send media session to disabled for a session agent that always redirects requests. It returns an error or 3xx response instead of forwarding an INVITE message. In addition, do not disable send media session on session agents that support SIP-to-H.323 IWF call flows. This can cause call failure.

22. response-map—Optional and for SIP only. Enter the name of the response map to use for this session agent. The mappings in each SIP response map is associated with a corresponding session agent. You can also configure this value for individual SIP interfaces.

23. ping-method—SIP only. Indicate the SIP message/method to use to ping a session agent. The ping confirms whether the session agent is in service. If this field is left empty, no session agent will be pinged.

    Setting this field value to the OPTIONS method might produce a lengthy response from certain session agents and could potentially cause performance degradation on your Oracle Enterprise Session Border Controller.

24. ping-interval—SIP only. Indicate how often you want to ping a session agent by configuring the ping interval parameter. Enter the number of seconds you want the Oracle Enterprise Session Border Controller to wait between pings to this session agent. The default value is 0. The valid range is:

    - Minimum: 0
    - Maximum: 999999999

    The Oracle Enterprise Session Border Controller only sends the ping if no SIP transactions (have occurred to/from the session agent within the time period you enter here.

25. trunk-group—Enter up to 500 trunk groups to use with this single session agent. Because of the high number of trunk groups you can enter, the ACLI provides enhanced editing mechanisms for this parameter:

    - You use a plus sign (+) to add single or multiple trunk groups to the session agent's list.

      When you add a single trunk group, simply use the plus sign (+) in front of the trunk group name and context. Do not use a Space between the plus sign and the trunk group name and context.

      For example, you might have already configured a list of trunk groups with the following entries: tgrpA:contextA, tgrpB:contextB, and tgrpC:contextC. To add tgrp1:context1, you would make the following entry:

      ```
      ACMEPACKET(session-agent)# trunk-group +tgrp1:context1
      ```

      Your list would then contain all four trunk groups.

      When you add multiple trunk groups, simply enclose your entry in quotation marks () or in parentheses (()). While you put spaces between the trunk group name and context entries, you do not use spaces with the plus sign, parentheses or quotation marks.

      ```
      ACMEPACKET(session-agent)# trunk-group +tgrp1:context1 tgrp2:context2
      tgrp3:context3
      ```

    - You use a minus sign (-) to delete single or multiple trunk groups from the session agent's list.

      When you remove a single trunk group, simply use the minus sign (-) in front of the trunk group name and context. Do not use a Space between the minus sign and the trunk group name and context.

      For example, you might have already configured a list of trunk groups with the following entries: tgrpA:contextA, tgrpB:contextB, tgrpC:contextC, and tgrp1:context1. To delete tgrp1:context1 from the list, you would make the following entry:

      ```
      ACMEPACKET(session-agent)# trunk-group -tgrp1:context1
      ```

      Your list would then contain: tgrpA:contextA, tgrpB:contextB, and tgrpC:contextC.

      When you add multiple trunk groups, simple enclose your entry in quotation marks () or in parentheses (()). While you put spaces between the trunk group name and context entries, you do not use spaces with the plus sign, parentheses or quotation marks.

      ```
      ACMEPACKET(session-agent)# trunk-group -tgrp1:context1 tgrp2:context2
      ```

    - You overwrite (replace) the entire list of a session agent's trunk groups by entering a list that does not use either the plus (+) or the minus (-) sign syntax.

26. ping-in-service-response-codes—SIP only. Enter the list of response codes that keep a session agent in service when they appear in its response to the Oracle Enterprise Session Border Controller's ping request. The Oracle Enterprise Session Border Controller takes the session agent out of service should a response code be used that does not appear on this list. Default is none.

27. out-service-response-codes—SIP only. Enter the list defines the response codes that take a session agent out of service when they appear in its response to the Oracle Enterprise Session Border Controller's ping request or any

in-dialog creating request (such as an INVITE, SUBSCRIBE, etc.). The Oracle Enterprise Session Border Controller ignores this list if an in-service list exists.

28. options—Optional. You can add your own features and/or parameters by using the options parameter. You enter a comma-separated list of either or both of the following:

    - feature=<value feature>

    For example:

    You can include the original address in the SIP message from the Oracle Enterprise Session Border Controller to the proxy in the Via header parameter by entering the following option:

    via-origin=<parameter-name>

    The original parameter is included in the Via of the requests sent to the session agent. The via origin feature can take a value that is the parameter name to include in the Via. If the value is not specified for via origin, the parameter name is origin.

    Note: If the feature value itself is a comma-separated list, enclose it within quotation marks.

29. media-profiles—Optional and for H.323 only. You can enter a list of media profiles to open logical channels when starting an outgoing call as a Fast Start H.323 call.

    Values you enter here must start with either an alphabetical character from A through Z (A-Za-z) or with an underscore (_). After the first character, each list entry can contain any combination of alphabetical or numerical characters (0-9A-Za-z), as well as the period (.), the dash (-), and the underscore (_). For example, netnet_mediaprofile1.

    You can enter 1 to 24 characters.

    Note: The values you enter here must correspond to a valid name you entered when you configure the media profile.

30. in-translationid—Optional. Enter the In Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to incoming traffic.

31. out-translationid—Optional. Enter the Out Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to outgoing traffic.

    Address translations attached to session agents take precedence over address translations attached to realms. If no address translation is applied to a session agent, then the Oracle Enterprise Session Border Controller will use the address translation applied to a realm. If an address translation is applied to both a realm and session agent, the translation attached to the session agent will apply. If the applicable session agent and realm have no associated translations, then the addresses will remain in their original forms and no address translations will be performed.

32. trust-me—Indicate whether this session agent is a trusted source, which the Oracle Enterprise Session Border Controller checks when it receives a message to determine if the source is trusted. The default value is disabled. The valid values are:

    - enabled | disabled

    The following example shows a session agent with an IP address used for the hostname.

```
session-agent
        hostname                        192.168.1.10
        ip-address                      192.168.1.10
        port                            5060
        state                           enabled
        app-protocol                    SIP
        app-type
        transport-method                UDP
        realm-id                        realm-1
        description                     englab
        carriers
                                        carrier1
        allow-next-hop-lp               enabled
```

```
        constraints                   disabled
        max-sessions                  355
max-inbound-sessions           4
        max-outbound-sessions         355
        max-burst-rate                0
        max-inbound-burst-rate        10
        max-outbound-burst-rate       1
        max-sustain-rate              3000
        max-inbound-sustain-rate      0
        max-outbound-sustain-rate     0
        min-seizures                  5
        min-asr                       0 time-to-resume                60
        ttr-no-response               0
        in-service-period             30
        burst-rate-window             60
        sustain-rate-window           3600
        req-uri-carrier-mode          None
        proxy-mode                    Proxy
        redirect-action               Recurse
        loose-routing                 enabled
        send-media-session            enabled
        response-map
        ping-method
        ping-interval                 0
        media-profiles
        in-translationid
        out-translationid
        trust-me                      disabled
        request-uri-headers
        stop-recurse
        local-response-map
        ping-to-user-part
        ping-from-user-part
        li-trust-me                   disabled
        in-manipulationid
        out-manipulationid
        p-asserted-id
        trunk-group
        max-register-sustain-rate     0
```

Session Agent Group Configuration

To configure session agent groups:

1. Access the **session-agent-group** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-group
ACMEPACKET(session-agent-group)#
```

2. **group-name**—Enter a unique name for the session agent group in Name format.
3. **description**—Optional. Enter descriptive information about the session agent group.
4. **state**—Enable or disable the session agent group on the Oracle Enterprise Session Border Controller. The default value is **enabled**. Valid values are:
   - enabled | disabled
5. **application-protocol**—Indicate the signaling protocol you want to use with the session agent group. The default value is **SIP**. The valid values are:
   - SIP | H.323

6. **strategy**—Indicate the session agent allocation strategy you want to use. The strategy you chose selects the session agents that will be made available by this session agent group. The default value is hunt. The valid values are:

   - **hunt**—Selects session agents in the order in which they are listed. For example, if the first agent is online, working, and has not exceeded defined constraints, all traffic is sent to the first agent. If the first agent is offline or if it exceeds a defined constraint, the second agent is selected. If the first and second agents are offline or exceed defined constraints, the third agent is selected. And so on through the list of session agents.
   - **roundrobin**—Selects each session agent in the order in which they are listed in the destination list, selecting each agent in turn, one per session.
   - **leastbusy**—Selects the session agent that has the fewest number of sessions relative to the maximum sessions constraint (for example, lowest percent busy) of the session agent element. The Least Busy Calculation is the result of dividing the number of active calls for a session agent by the max-sessions parameter within the session-agent element configuration. If the default max-session parameter value issued for a session agent (0), the result of the Least Busy Calculation will be 0. The Least Busy SAG Strategy will route a session to the session agent with the lowest resulting Least Busy Calculation percentage. If multiple session agents have the lowest percentage, the foremost session agent in the Session Agent Group dest parameter will be used.
   - **propdist**—Based on programmed, constrained session limits, the Proportional Distribution strategy proportionally distributes the traffic among all of the available session agents. Sessions are distributed among session agents based on the max-outbound-sessions value in each session agent. The sum of max-outbound-sessions for every session-agent within a session group equates to 100% and the max-outbound-sessions value for each session-agent represents a % that total. Sessions are proportionally allocated to session agents based on their individual session agent max-outbound-sessions value, as a % of the total max-outbound-sessions for the group.
   - **lowsusrate**—The Lowest Sustained Rate strategy routes to the session agent with the lowest sustained rate of session initiations/invitations (based on observed sustained session request rate).
7. **destination**—Identify the destinations (session agents) available for use by this session agent group.

   A value you enter here must be a valid IP address or hostname for a configured session agent.
8. **trunk-group**—Enter trunk group names and trunk group contexts to match in either IPTEL or custom format. If left blank, the Oracle Enterprise Session Border Controller uses the trunk group in the realm for this session agent group. Multiple entries are surrounded in parentheses and separated from each other with spaces.

   Entries for this list must one of the following formats: trgp:context or trgp.context.
9. **sag-recursion**—Enable this parameter if you want to use SIP SAG recursion for this SAG. The default value is disabled. Valid values are:

   - enabled | disabled
10. **stop-sag-recurse**—Enter the list of SIP response codes that terminate recursion within the SAG. Upon receiving one of the specified response codes, such as 401 unauthorized, or upon generating one of the specified response codes internally, such as 408 timeout, the Oracle Enterprise Session Border Controller returns a final response to the UAC. You can enter the response codes as a comma-separated list or as response code ranges.
11. Type **done** to save your configuration.

SAG Matching for LRT and ENUM

When this feature is enabled and a match is found, the Oracle Enterprise Session Border Controller uses the matching SAG for routing. When there is no match for the SAG, the Oracle Enterprise Session Border Controller processes the result as it would have if this feature had not been enabled: either matching to a session agent hostname, or performing a DNS query to resolve it.

Note that you set the state of this feature in the SIP configuration.

To configure a SAG for ENUM or LRT matching:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   If you are adding support for this feature to a pre-existing SIP configuration, then you must select (using the ACLI select command) that configuration to edit it.

4. enum-sag-match—Set this parameter to enabled so the Oracle Enterprise Session Border Controller will match session agent group (SAG) names with the hostname portion in the naming authority pointer (NAPTR) from an ENUM query or LRT next-hop entry. The default value is disabled. The valid values are:

   - enabled | disabled

5. Save and activate your configuration.

Configuring Local Policy

To configure local policy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type local-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# local-policy
   ACMEPACKET(local-policy)#
   ```

4. from-address—Indicate the originating address information by entering a From address value. You can use the asterisk (*) as a wildcard to indicate this policy can be used with all originating addresses.

   You can also use complete or partial E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:

   - SIP From address
   - FQDNs
   - IP addresses
   - H.323 CallingPartyAddress

     The Oracle Enterprise Session Border Controller also supports the asterisk as part of the From address you configure in your local policies.

     This means that for the from-address parameters of a local policy configuration, you can enter values in which an asterisk appears and match them accordingly. You might enter a value that resemble the following example:
   - 123*456

   Note: After entering the from-address value, the Oracle Communications Session Delivery Manager automatically saves it to the configuration when exiting from local policy.

5. to-address—Indicate the destination address by entering a To address value. You can use the asterisk (*) as a wildcard to indicate all this policy can be used for any destination address.

   You can also use E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:

   - SIP Request-URI
   - FQDNs
   - IP addresses
   - H.323 CalledPartyAddress

The system also supports the asterisk as part of the To address you configure in your local policies.

This means that for the to-address parameters of a local policy configuration, you can enter values in which an asterisk appears and match them accordingly. You might enter a value that resembles the following example:

- 123*456

Note: After entering the to-address value, the Oracle Communications Session Delivery Manager automatically saves it to the configuration when exiting from local policy.

6. source-realm—Enter the realm, or list of realms, you want the Oracle Enterprise Session Border Controller to use to determine how to route traffic. This list identifies from what realm traffic is coming and is used for routing by ingress realm by the local policy.

   You can use the asterisk (*) as a wildcard to indicate this local policy can be used with all realms. The default value is *.Or you can enter a value that corresponds to the identifier of an already configured realm. Formats include the following:

   - realm ID
   - customer name
   - peer name
   - subdomain name
   - VPN identifier

7. activate-time—Set the time you want the local policy to be activated using the following syntax:

   ```
   yyyy:mm:dd hh:mm:ss
   yyyy:mm:dd-hh:mm:ss
   ```

8. deactivate-time—Set the time you want the local policy to be deactivated using the following syntax:

   ```
   yyyy:mm:dd hh:mm:ss
   yyyy:mm:dd-hh:mm:ss
   ```

9. state—Indicate whether you want the local policy to be enabled or disabled on the system. The default value is enabled. The valid values are:

   - enabled | disabled

10. policy-attribute—Configure local policy attributes by following steps 8 through 21.
11. next-hop—Identify the next signaling host by entering the next hop value. You can use the following as next hops:

    - IPv4 address or IPv6 address of a specific endpoint
    - Hostname or IPv4 address or IPv6 address of a configured session agent
    - Group name of a configured session agent group

      Note: The group name of a configured session agent group must be prefixed with SAG:

      For example:

      policy-attribute

      next-hop SAG:appserver

      policy-attribute

      next-hop lrt:routetable

      policy-attribute

      next-hop enum:lerg

      You can also configure a next hop that has an address of 0.0.0.0, thereby creating a null route. Different from not having a local policy configured (which would trigger Oracle Enterprise Session Border Controller local policy recursion), this terminates local policy recursion and immediately fails the request. In these cases, the system responds a request with a 404 Not Found.

12. realm—Identify the egress realm (the realm used to reach the next hop) if the system must send requests out from a specific realm.

The value you enter here must correspond to a valid identifier you enter when you configured the realm. If you do not enter a value here, and the next hop is a session agent, the realm identified in the session agent configuration is used for egress. In H.323, the next hop address is matched against the realm's address prefix to determine the realm.

13. replace-uri—Indicate whether you want to replace the Request-URI in outgoing SIP requests with the next hop value.
14. carrier—Optional. Enter the name of the carrier associated with this route. The value you enter here must match one or more of the carrier names in the session agent configuration.

    Entries in carrier fields can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! " # $ % ^ & * ( ) + - = < > ? ' | { } [ ] @ / \ ' ~ , . _ : ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carrier are valid carrier field formats.
15. start-time—Indicate the time of day (from the exact minute specified) the local policy attributes go into effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

    ```
    1400
    ```

    The default value of 0000 implies that the defined policy attributes can be considered in effect any time after 00:00:00. The valid range is:

    - Minimum—0000
    - Maximum—2400
16. end-time—Indicate the time of day (from the exact minute specified) the local policy attributes are no longer in effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

    ```
    2400
    ```

    The default value of 2400 implies that the defined policy attributes can be considered in effect any time before midnight. The valid range is:

    - Minimum—0000
    - Maximum—2400
17. days-of-week—Enter any combination of days of the week (plus holidays) you want the local policy attributes to be in effect. You must enter at least one day or holiday here. A holiday entry must correspond with a configured holiday established in the Session Router.

    The default is U-S. The valid values are:

    - U (Sunday)
    - M (Monday)
    - T (Tuesday(
    - W (Wednesday)
    - R (Thursday)
    - F (Friday)
    - S (Saturday)
    - H (Holiday)

      You can enter a range of values separated by a hyphen, for example U-S. And you can enter multiple values separated by commas, for example M,W,F. You cannot use spaces as separators.
18. cost—Enter a cost value that acts as a unitless representation of the cost of a route relative to other routes reaching the same destination (To address). This value is used as a way of ranking policy attributes.

    The default value is zero (0). The valid values are:

    - minimum—zero (0)
    - maximum—999999999
19. app-protocol—Enter the signaling protocol to use when sending messages to the next hop. The valid values are:

    - H.323 | SIP

20. state—Indicate whether you want to enable or disable the local policy. The default value is enabled. The valid values are:

    - enabled | disabled

21. media-profiles—Configure a list of media profiles if you want the local policy to route SIP and H.323 traffic by the codecs specified in the SDP. The list of media profiles entered here are matched against the SDP included in SIP or H.323 requests and the next hop is selected by codec.

    The values in this list are matched against the rtpmap attribute of passed SDP, and preference weight for route selection is based on the order in which the matching payload type appears in the SDP's media (m=) line.

    For example when the following SDP arrives:

    ```
    m=audio 1234 RTP/AVP 0 8 18
    ```

    that contains the following attributes that correspond to three configured local policies with the same cost:

    - a=rtpmap:0 PCMU/8000
    - a=rtpmap:8 PCMA/8000
    - a=rtpmap:18 G729/8000

    the following route selection action occurs:

    The local policy route that corresponds to the a=rtpmap:0 PCMU/8000 attribute is selected because the payload type of 0 in the attribute line matches the first payload type of 0 listed in the m= line. The codec value of PCMU indicated in this selected attribute is used to find the local policy with the media profiles attribute that includes PCMU in the list.

    Because the value you enter here is matched against the codec values included in the actual passed SDP, it must correspond to accepted industry-standard codec values.

    The following example shows a local policy with a next hop value of the session agent group called gw-sag2.

```
local-policy
        from-address
                                        *
        to-address
                                        192.168.1.10
        source-realm                    *
        activate-time                   2005-01-20 20:30:00
        deactivate-time                 N/A
        state                           enabled
        last-modified-date              2005-01-10 00:36:29
policy-attribute
        next-hop                        SAG:gw-sag2
        realm
        replace-uri                     enabled
        carrier
        start-time                      0000
        end-time                        2400
        days-of-week                    U-S
        cost                            0
        app-protocol
        state                           enabled
        media-profiles
```

Local Policy Matching for Parent Realms

For SIP and H.323, you can configure the Oracle Enterprise Session Border Controller to use the parent realm for routing purposes even when the source realm for an incoming message is a child realm.

With this feature disabled (default), the Oracle Enterprise Session Border Controller uses the specific source realm to perform a local policy look-up. When the source realm is a child realm and any relevant local policies are configured with the parent realm, there will be no matches and the local policy look-up will fail. To avoid this issue and ensure successful look-ups, you must configure multiple local policies if you want to use a configuration with nested realms.

The Oracle Enterprise Session Border Controller examines the source realm to determine if it is a parent realm with any child realms when you enable this feature. If the parent, source realm does have child realms, then the Oracle Enterprise Session Border Controller creates local policy entries for the parent and all of its child realms. This operation is transparent and can save time during the configuration process.

It is possible, then, for a local policy look-up to match the same child realm in two ways:

- Through a match via the parent realm
- Through a direct match for a local policy configured with that specific child realm

In such a case, the child realm must have different costs for each type of match to avoid collisions.

This feature is enabled on a global basis in the session router configuration. Because it applies system-wide, all source realms will use this form of matching when enabled.

To enable local policy matching for parent realms:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-router and press Enter.

   ```
   ACMEPACKET(session-router)# session-router
   ACMEPACKET(session-router-config)#
   ```

4. match-lp-source-parent-realms—If you want the Oracle Enterprise Session Border Controller to perform local policy realm matching based on the parent realm (so that there are local policy entries for parent and child realms), set this parameter to enabled. The default value is disabled. The valid values are:

   - enabled | disabled

   ```
   ACMEPACKET(session-router-config)# match-lp-src-parent-realms enabled
   ```

5. Save and activate your configuration.

SIP Session Agent DNS-SRV Load Balancing

Prior to Release 6.2.0 the Oracle Enterprise Session Border Controller provided the ability to specify an FQDN (fully qualified domain name) for a destination session-agent. During DNS lookup the FQDN could resolve to multiple SRV (Resource Record for Servers) records. Each SRV could resolve to a single IP address via A-Record query in IMS or DNS.

With Release 6.2.0 the Oracle Enterprise Session Border Controller supports load balancing behavior as described in RFC 3263, Session Initiation Protocol (SIP): Locating SIP Servers.

The Oracle Enterprise Session Border Controller will provide a new parameter ping-all-addresses in session-agent configuration mode to enable internal load balancing and RFC 3263 compliance. The Oracle Enterprise Session Border Controller monitor the availability of the dynamically resolved IP addresses obtained from DNS server using OPTIONS ping (ping-per-DNS entry). The ping-method and ping-interval for each resolved IP addresses will be copied from original session-agent.

Status of Session-Agent:

In Service – if any of dynamically resolved IP addresses is in service

Out of service – if all dynamically resolved IP addresses is out of service.

The default of ping-all-addresses is disabled, in which case the Oracle Enterprise Session Border Controller only pings the first available resolved IP addresses.

With status of each resolved IP addresses above, the Oracle Enterprise Session Border Controller will recurse through the list of these in-service IP addresses dynamically resolved from DNS server on 503 response, and stop recursion based upon a configured list of response values specified by the stop-recurse parameter in sip-interface configuration

mode. With internal load balancing enabled in the session-agent, the Oracle Enterprise Session Border Controller provides the ability to select routing destinations based on SRV weights. The priority/weight algorithm is based on RFC 2782, *A DNS RR for specifying the location of services (DNS SRV)*.

The Oracle Enterprise Session Border Controller will provide the similar functionality as that listed above for A-records, the Oracle Enterprise Session Border Controllerwill select first available routing destinations because there is no priority/weight contained in A-records.

Session Agent DNS-SRV Load Balancing Configuration

To configure the Oracle Enterprise Session Border Controller to perform Session-Agent DNS-SRV load balancing:

1. From superuser mode, use the following command sequence to access sip-config configuration mode. While in this mode, you configure SAG-based address resolution.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

2. Use the ping-all-addresses parameter to enable Session-Agent DNS-SRV load balancing.
3. Use done, exit, and verify-config to complete Session-Agent DNS-SRV load balancing configuration.

   The show agents ACLI command displays the availability of dynamically resolved IP addresses

```
ACMEPACKET# show sip agents acme.engr.com
21:46:05-51-router
Session Agent acme.engr.com(core) [In Service] NO ACTIVITY
Session Agent acme.hxu.com(core) [In Service] NO ACTIVITY
Destination: 192.168.200.235 In Service
Destination: 192.168.200.231 In Service
...
...
```

Answer to Seizure Ratio-Based Routing

New SIP session agent constraints set a threshold for Answer to Seizure Ratio (ASR) has been implemented. ASR is considered when determining whether session agents are within their constraints to route calls (in addition to session and rate constraints).

The new session agent constraints indicate the minimum acceptable ASR value and computes the ASR while making routing decisions. ASR is calculated by taking the number of successfully answered calls and dividing by the total number of calls attempted (which are known as seizures).

If the ASR constraints are exceeded, the session agent goes out of service for a configurable period of time and all traffic is routed to a secondary route defined in the local policy (next hop with higher cost).

The two session agent constraints are:

- minimum seizure: determines if the session agent is within its constraints. When the first call is made to the session agent or the if calls to the session agent are not answered, the minimum seizure value is checked.

  For example, if 5 seizures have been made to the session agent and none of them have been answered, the sixth time, the session agent is marked as having exceeded its constraints and the calls will not be routed to it until the time-to-resume has elapsed.
- minimum ASR: considered when make routing decisions. If some or all of the calls to the session agent have been answered, the minimum ASR value is considered to make the routing decisions.

  For example, if the you set the minimum ASR at 50% and the session agent's ASR for the current window falls below 50%, the session agent is marked as having exceeded its constraints and calls will not be routed to it until the time-to-resume has elapsed.

ASR Constraints Configuration

To configure ASR constraints:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. If configuring an existing session agent, enter the select command to select the session agent.
5. min-seizures—Enter the minimum number of seizures that when exceeded, cause the session agent to be marked as having exceeded its constraints. Calls will not be routed to the session agent until the time-to-resume has elapsed. The default value is 5. The valid range is:
   - Minimum—1
   - Maximum—999999999
6. min-asr—Enter the percentage you want as the minimum. If the session agent's ASR for the current window falls below this percentage, the session agent is marked as having exceeded its constraints and calls will not be routed to it until the time-to-resume has elapsed. The default value is 0. The valid range is:
   - Minimum—0
   - Maximum—100
7. Save and activate your configuration.

   The following example shows a session agent configuration.

```
session-agent
        hostname                        192.168.1.6
        ip-address
        port                            1720
        state                           enabled
        app-protocol                    H323
        app-type                        H323-GW
        transport-method
        realm-id                        external
        description
        carriers
        allow-next-hop-lp               enabled
        constraints                     enabled
        max-sessions                    0
max-inbound-sessions            4
        max-outbound-sessions           5
        max-burst-rate                  0
        max-inbound-burst-rate          10
        max-outbound-burst-rate         1
        max-sustain-rate                0
        max-inbound-sustain-rate        0
        max-outbound-sustain-rate       0
min-seizures                    5
        min-asr                         50
        time-to-resume                  30
        ttr-no-response                 0
        in-service-period               0
        burst-rate-window               0
        sustain-rate-window             0
        req-uri-carrier-mode            None
        proxy-mode
```

```
        redirect-action
        loose-routing                   enabled
        send-media-session              enabled
        response-map
        ping-method
        ping-interval                   0
        media-profiles
        in-translationid
        out-translationid
        trust-me                        disabled
        request-uri-headers
        stop-recurse
        local-response-map
        ping-to-user-part
        ping-from-user-part
        li-trust-me                     disabled
        in-manipulationid
        out-manipulationid
        p-asserted-id
        trunk-group
        max-register-sustain-rate       0
        early-media-allow
        invalidate-registrations        disabled
        last-modified-date              2006-05-12 19:48:06
```

Active Directory-based Call Routing

A large percentage of Enterprises currently use call servers with Active Directory (Domain Controller) such as Media Servers, Exchange Servers, Lync Servers, etc. For Enterprises that integrate these servers in parallel to their existing communications infrastructure, or transition from their legacy Private Branch Exchange (PBX) to these types of servers, Active Directory becomes a more efficient and cost-effective way of routing the incoming calls within the core Enterprise network.

Clients using Microsoft servers such as a Lync Server deploy their own URI. Therefore, a user in a network with both a desk phone and a Lync client have an IP PBX extension/URI for the desk phone, and a different URI for the Lync client. Currently, all PSTN traffic is sent by default, to a legacy PBX in the core network. If the PBX does not recognize the extension/URI, the PBX forwards it to the Lync client. Sending traffic to the PBX first and then to the Lync Server can be costly in terms of compute resources and/or licensing fees. Routing all incoming sessions from a SIP trunk to the Lync Server first and then to a PBX can be costly.

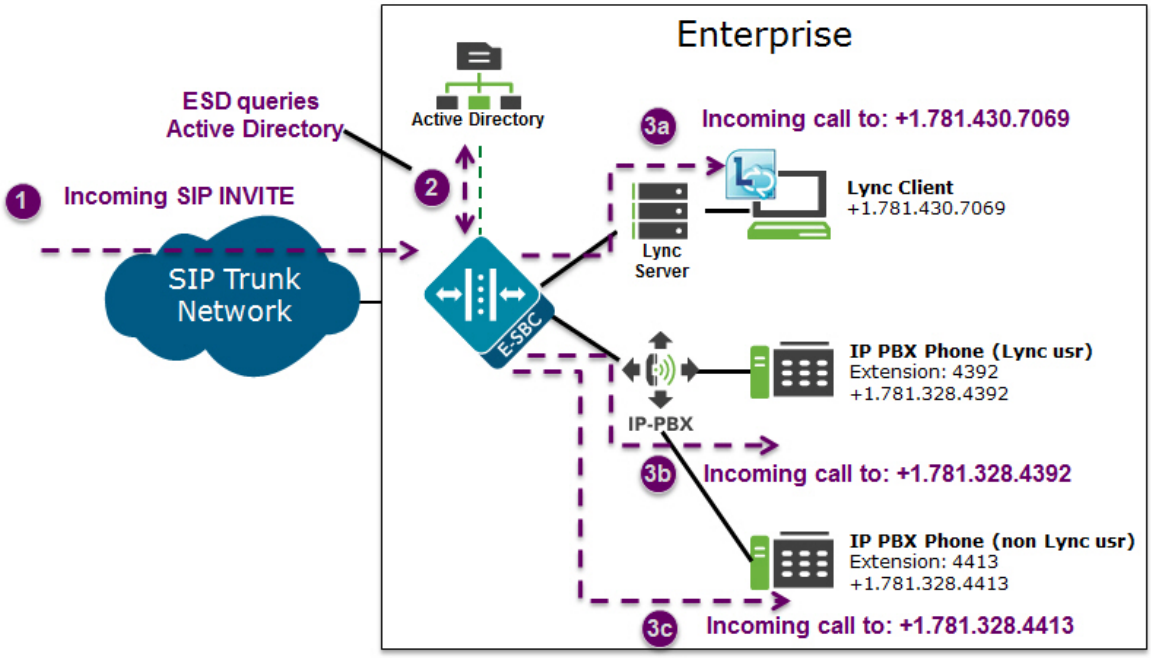
As a solution, the Oracle Enterprise Session Border Controller initiates a query to the Active Directory to initially determine the type of incoming call. The Oracle Enterprise Session Border Controller then stores data used to facilitate the routing decision of the call (performed by Lightweight Directory Access Protocol (LDAP)) and then routes the call the first time to the applicable destination (PBX or Lync Server).

In scenarios where a user has both a Lync phone and a legacy PBX phone, calls destined for the Lync phone number can be routed to the PBX phone number, or calls destined for the PBX phone number can be routed to the Lync phone number. The destination is dependant on the current Oracle Enterprise Session Border Controller configuration. The Oracle Enterprise Session Border Controller uses the information stored in the Enterprise's Active Directory, compares it to the ESD configuration and then determines which phone number to utilize for the destined user.

Note: The Active Directory-based call routing feature supports confidential and secure LDAP traffic support by using SSL/TLS (LDAPS).

Active Directory-based call routing is a feature of the Oracle Enterprise Session Border Controller that facilitates the routing of incoming calls to the appropriate destinations within the Enterprise core network. The Oracle Enterprise Session Border Controller's LDAP query to the Active Directory yields whether or not the phone number is associated with a call Server or the PBX.

When the Oracle Enterprise Session Border Controller receives an inbound SIP INVITE over a SIP Trunk (1), it checks the current LDAP configuration in the Oracle Enterprise Session Border Controller. Depending on this configuration, the Oracle Enterprise Session Border Controller then accesses the Enterprise's Active Directory to search for the applicable number being called via an LDAP query (2). When the query has found the number to forward the call, the Oracle Enterprise Session Border Controller routes the call directly to the call server client (3a) or to the IP PBX phone (3b) and 3c) as shown in the illustration below.



The Enterprise is responsible for migrating phone numbers from the legacy PBX to the call server by making the necessary updates in their Active Directory in order for the Oracle Enterprise Session Border Controller to route the call properly. In the illustration above, the IP PBX extension (4392) is the primary telephone number (+1.781.328.4392); a secondary transition number (+1.781.430.7069) is assigned to Lync.

LDAP in the Oracle Enterprise Session Border Controller

Lightweight Directory Access Protocol (LDAP) is the Protocol that the Oracle Enterprise Session Border Controller uses to perform queries to the Enterprise's Active Directory to determine where to route incoming calls (to the call server or the IP PBX) in the Enterprise network. Session requests and responses are sent/received based on the Oracle Enterprise Session Border Controller's LDAP routing configuration. LDAP determines the destination (call server user or non-call server user) and forwards the call accordingly.

The Oracle Enterprise Session Border Controller, using LDAP, performs the following on an inbound call:

- Creates an LDAP search filter based on the dialed number and the configured LDAP attributes.
- Sends an LDAP search query to the configured LDAP Server.
- Creates a route list based on the query response received from the LDAP Server.
- Routes calls to both the call server and the IP PBX. The routing order is dependent on the LDAP attribute configuration and/or whether there was an exact match for the dialed phone number in the Enterprise's Active Directory for the call server or the IP-PBX.

Note: You configure LDAP Servers, filters, and local policy routing using the ACLI objects and attributes. For more information about configuring LDAP, see *Configuring LDAP*.

The Oracle Enterprise Session Border Controller keeps a permanent LDAP session open to all configured call servers. It sends an LDAP bind request on all established connections, to those servers. The first call server is considered the primary LDAP Server, and all others are secondary LDAP servers. If a query request sent to the primary server fails,

the Oracle Enterprise Session Border Controller sends the request to the next configured LDAP Server, until the request is successful in getting a response. If no response is received by the Oracle Enterprise Session Border Controller and the Oracle Enterprise Session Border Controller cannot find another route successfully, (all Oracle Enterprise Session Border Controller configured attributes have been exhausted (local policies, policy attributes, etc.)), it sends a busy to the caller.

LDAP performs call routing based on LDAP attributes configured on the Oracle Enterprise Session Border Controller. The route-mode attribute setting determines how LDAP handles the called number when accessing the Enterprise's Active Directory. Routing modes can be set to any of the following:

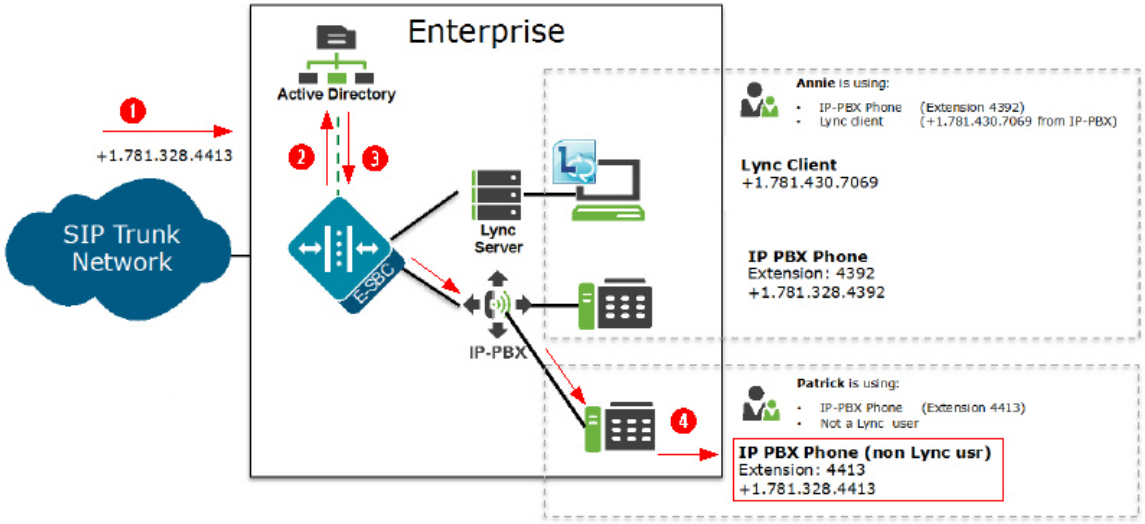
- Exact-match-only (default)
- Attribute-order-only
- Exact-match-first

The following paragraphs describe each of these route-modes.

Exact-match-only

If the LDAP route-mode attribute is set to exact-match-only, the Oracle Enterprise Session Border Controller performs as follows.

The Oracle Enterprise Session Border Controller receives an incoming call to the Enterprise network. If the LDAP route-mode attribute on the Oracle Enterprise Session Border Controller is set to exact-match-only, LDAP queries the Active Directory to find the number that matches exactly to the incoming number. If the number is found, the Oracle Enterprise Session Border Controller forwards the call to the client's applicable phone in the Enterprise network.



Number	Description
①	Call comes into the Enterprise network (+1.781.328.4413)
②	Using the configured route-mode of exact-match-only, LDAP queries the exact matching number in the Enterprise's Active Directory.
③	The Active Directory finds the matching number and that number is included in the response to the LDAP query.
④	The Oracle Enterprise Session Border Controller forwards the call to the destination phone number (same number as the number that initially called into the Enterprise in Step 1 (+1.781.328.4413)).

Attribute-order-only

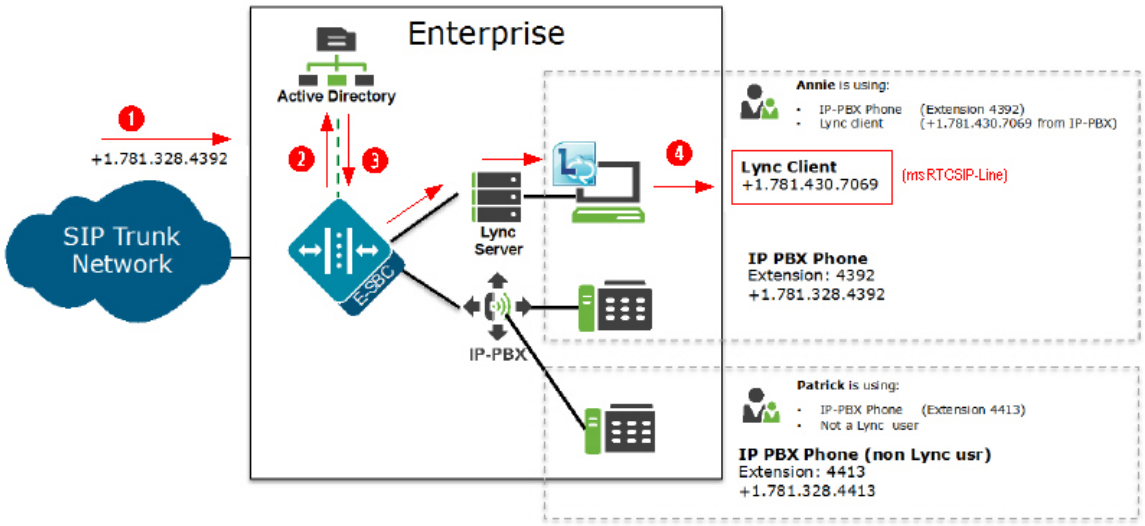
If the LDAP route-mode attribute is set to attribute-order-only, the Oracle Enterprise Session Border Controller performs as follows.

The order in which the LDAP attributes are configured on the Oracle Enterprise Session Border Controller determines the priority of each route. If an incoming call is destined for the IP-PBX , but the attribute name for a Lync client is configured first, the Oracle Enterprise Session Border Controller uses the corresponding next hop (Lync Server) to create the first route in the route list.

An entry in an LDAP search response must have at least one attribute that it matches in the Active Directory.

For example, the incoming phone number could be +1.781.328.4392 (which matches the IP-PBX phone number), and the attribute name msRTCSIP-Line (Lync attribute) in the response could be +1.781.430.7069 (Lync phone number). A route is created for the Lync phone number, even though the incoming phone number matches the IP-PBX phone number, since the msRTCSIP-Line attribute was configured first. Therefore, the Oracle Enterprise Session Border Controller forwards the call to the Lync destination.

Likewise, if an Enterprise uses the same phone number for both Lync and IP-PBX phones, and the attribute-name msRTCSIP-Line is configured first (a Lync attribute), the Oracle Enterprise Session Border Controller uses the corresponding next hop (Lync Server) to create the first route in the route list.



Number	Description
①	Call comes into the Enterprise network (+1.781.328.4392)
②	Using the configured route-mode of attribute-order-only, LDAP queries the Active Directory for the matching number.
③	The Active Directory responds with the phone number associated with the first configured LDAP attribute (+1.781.430.7069). In the illustration above, the number was associated with a Lync Client (msRTCSIP-Line) that was configured first in the LDAP configuration.
④	The Oracle Enterprise Session Border Controller forwards the call to the applicable destination phone number from the Active Directory response. (+1.781.430.7069).

If you configure the attribute name msRTCSIP-Line first, the Oracle Enterprise Session Border Controller uses the corresponding next hop (Lync Server) to create the second highest priority route in the route list. For example, the dialed telephone number could be +1.781.328.4392 (IP-PBX phone number), and the attribute-name msRTCSIP-Line

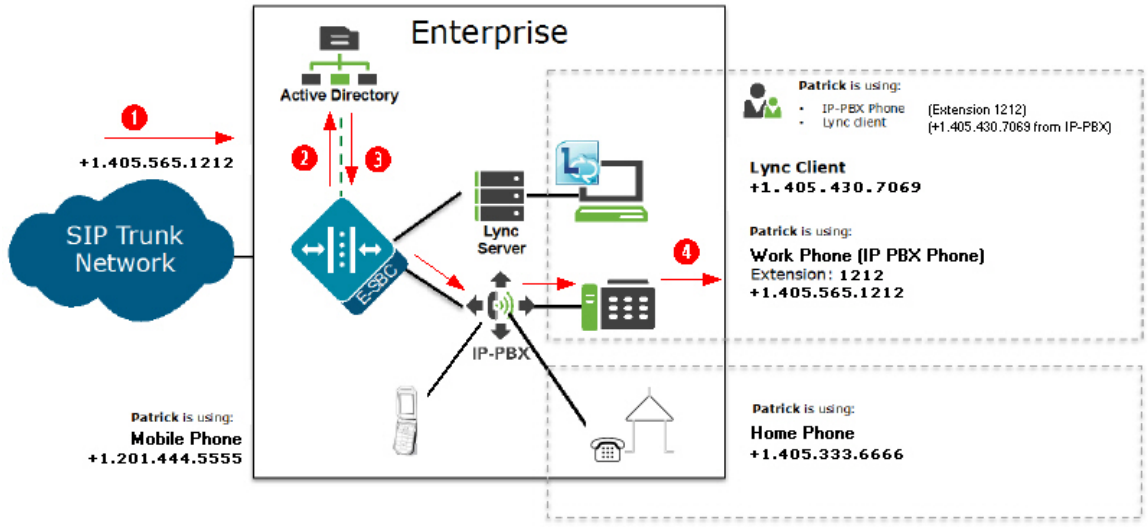
in the response could be +1.781.430.7069 (Lync phone number). A route is created for the Lync phone number, even though the dialed telephone number is the PBX phone number.

Exact-match-first

If the LDAP route-mode attribute is set to exact-match-first, the Oracle Enterprise Session Border Controller performs as follows.

When the LDAP query is sent to the Active Directory, the first exact match of the incoming phone number that the LDAP query finds in the Directory, is the number whose corresponding route gets the highest priority in the route list. For all other routes configured on the Oracle Enterprise Session Border Controller, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route.

For example, if the incoming number is +1.405.565.1212, and the Active Directory includes a configured mobile number first (+1.201.444.5555), a home number second (+1.405.333.6666) , and a work number third(+1.405.565.1212), the LDAP query searches the mobile number first, then the home number, then finds the exact match on the work phone number. The Active Directory responds with the destination information for the work phone number and the Oracle Enterprise Session Border Controller creates a route list with this exact phone number, and then forwards the call accordingly.



Number	Description
①	Call comes into the Enterprise network (+1.405.565.1212)
②	Using the configured route-mode of exact-match-first, LDAP queries the Active Directory for the matching number.
③	The LDAP query searches throughout the Active Directory until it finds the first exact match on the number. Active Directory responds with the exact phone number associated with the incoming number (+1.405.565.1212). In the illustration above, the number was associated with the work phone.
④	The Oracle Enterprise Session Border Controller forwards the call to the applicable destination phone number from the Active Directory response. (+1.405.565.1212).

LDAP Messages

If LDAP message logging is enabled in the Active Directory, the Oracle Enterprise Session Border Controller sends LDAP messages to a message log called sipdldap.log. This log records all received and sent LDAP messages. Messages are in ASCII encoded binary format.

Additionally, when LDAP is invoked for routing, the LDAP messages display in the GUI under the Monitor and Trace tab. For more information about viewing LDAP messages in the GUI, see the *Net-Net Enterprise Session Director Web GUI User Guide*.

Note: The Oracle Enterprise Session Border Controller also supports transmitting LDAP messages using the IPFIX Protocol for the Palladion Mediation Engine.

LDAP Failure Events

If an incoming session to a primary phone number routed to Lync fails, the phone number is routed to the IP PBX. If failures occur during LDAP queries for all LDAP Servers, the Oracle Enterprise Session Border Controller logs the failure to the sipdldap.log, and proceeds with normal configured routing policies, if available.

Note: The Oracle Enterprise Session Border Controller always establishes the TCP/TLS connection towards the configured LDAP server(s). If a TCP connection fails, the Oracle Enterprise Session Border Controller continues to attempt to re-establish the connection.

An LDAP connection failure can be due to any one of the following events:

- Oracle Enterprise Session Border Controller receives a CANCEL message (LDAP connection termination). The Oracle Enterprise Session Border Controller detects this if it receives or issues an 'unbind' operation. The session is then closed down at TCP/TLS.
- Oracle Enterprise Session Border Controller receives a call failure message from Lync (TCP/TLS socket termination). If either side receives a finish message (FIN) or reset message (RST), the TCP socket closes per standard behavior, which triggers the LDAP layer to detect connection failure. The Oracle Enterprise Session Border Controller fails over to a secondary LDAP Server, if configured; otherwise it periodically attempts to reconnect to the Primary LDAP Server.
- Oracle Enterprise Session Border Controller is unreachable and SIP session towards Lync times out. User is enabled for Lync but the Lync Server is unreachable by the Oracle Enterprise Session Border Controller so a timeout occurs. When consecutive LDAP queries timeout, the Oracle Enterprise Session Border Controller concludes that the LDAP session has failed, and then proceeds to terminate the TCP/TLS connection.

The number of consecutive queries that timeout before a connection is considered failed, and the number of successive query timeouts for each LDAP Server can be set via configuration.

Oracle Enterprise Session Border Controller Limitations using LDAP

The Oracle Enterprise Session Border Controller uses LDAP in the Active Directory when determining the destination of incoming calls. However, the Oracle Enterprise Session Border Controller has the following limitations when using LDAP:

- Supports LDAP sessions over the Oracle Enterprise Session Border Controller media interfaces only (i.e., not on wancom0).
- Supports LDAPv3 only.
- Establishes a session over the following connections only:

  LDAP over TCP - default

  LDAP over TLS (LDAPS)

Configuring LDAP

LDAP is the Protocol that the Active Directory uses for general interaction between and LDAP client and an LDAP server. You can configure the LDAP Server(s) in your network, and set the filters and the local policy that the LDAP Server uses when handling inbound Lync and PBX calls in the Enterprise core network.

You can use the following objects in the ACLI to configure LDAP.

Object	XML Tag	ACLI Path	Description
ldap-config	ldapConfig	session-router->ldap-config	Configures the LDAP functionality on the Oracle Enterprise Session Border Controller (i.e., name, state, LDAP servers, realm, authentication mode, username, password, LDAP search filters, timeout limits, request timeouts, TCP keepalive, LDAP security type, LDAP TLS profile, and LDAP transactions). Note: This is a multiple-instance object.
ldap-transaction	ldapTransaction	session-router->ldap-config-> ldap-transaction	Configures the application transaction type for LDAP, determines route priority in the route list, and configures the LDAP configuration attributes. You configure this object for LDAP search queries in call routing. Note: This is a multiple-instance object.
ldap-cfg-attributes	ldapCfgAttributes	session-router->ldap-config-> ldap-transaction->ldap-cfg-attributes	Configures the Active Directory attribute name, next hop for routing SIP requests, the realm for the next hop, a regular expression pattern, and a format for the attribute value. You configure this object for LDAP search queries in the Active Directory. Note: This is a multiple-instance object.
policy-attributes	policyAttributes	session-router->local-policy-> policy-attributes	Configures the ldap: prefix with the name of the ldap-config. This allows the Oracle Enterprise Session Border Controller to send LDAP queries to the Active Directory server(s) configured in the ldap-config element whenever there is a match for the corresponding local-policy. Note: An ldap-config with the LDAP name specified for this parameter must be configured for the next hop. An LDAP next hop is supported only for SIP to SIP calls. This is a multiple-instance object.

Configuring ldap-config

You use the ldap-config object in the ACLI to create and enable an LDAP configuration on the ESD.

To configure ldap-config:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type ldap-config and press Enter to access the LDAP configuration-related attributes.

```
ACMEPACKET(session-router)# ldap-config
ACMEPACKET(ldap-config)#
```

- name—Enter a name to assign to this LDAP configuration. This is a unique identifier. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: name

  ```
  ACMEPACKET(ldap-config)# name ldapquery
  ```
- state—Specify whether or not to enable the operational state of the LDAP configuration. When the state is disabled, ESD does not attempt to establish any connection with the corresponding LDAP Server(s). Default is enabled. Valid values are:

  - enabled (default)
  - disabled

  XML Tag: state

  ```
  ACMEPACKET(ldap-config)# state enabled
  ```
- ldap-servers—Enter the IP address(es) and optionally the port number(s) for each LDAP Server(s) you want to add to the LDAP configuration. When more than one server is specified, each server address should be separated by a space and the list enclosed within parentheses. The first server listed is considered the primary LDAP Server, and the remaining servers are considered the secondary LDAP Servers. The HUNT strategy is used to determine the active LDAP Server (where the ESD selects the first LDAP Server; if unreachable, it selects the second LDAP Server; it that is unreachable, it selects the third LDAP Server, etc.). Default ports used are 389 (for LDAP over TCP) and 636 (LDAP over TLS). IP Address must be entered in dotted decimal format (0.0.0.0). Default is blank.

  XML Tag: ldapServers

  ```
  ACMEPACKET(ldap-config)# ldap-servers (172.44.0.20:636 172.44.0.21:389)
  ```
- realm—Enter the name of the realm that determines which network interface to issue an LDAP query. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: realm

  ```
  ACMEPACKET(ldap-config)# realm net172
  ```
- authentication-mode—Specify the authentication mode to use in the LDAP bind request. Default is Simple. No specific password encryption is done when sending the bind request. You can use an LDAPS connection with the LDAP Server to maintain security (see ldap-sec-type).

  XML Tag: authType

  ```
  ACMEPACKET(ldap-config)# authentication-mode Simple
  ```
- username—Enter the username that the LDAP bind request uses for authentication before access is granted to the LDAP Server. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: username

  ```
  ACMEPACKET(ldap-config)# username ENGLAB\Administrator
  ```
- password—Enter the password to be paired with the username attribute, that the LDAP bind request uses for authentication before access is granted to the LDAP Server. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: password

  ```
  ACMEPACKET(ldap-config)# password sips1234
  ```
- ldap-search-base—Enter the base Directory Number you can use for LDAP search requests. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: ldapSearchBase

  ```
  ACMEPACKET(ldap-config)# ldap-search-base
  CN=Users,DC=englab,DC=acmepacket,DC=com
  ```
- timeout-limit—Enter the maximum amount of time, in seconds, for which the ESD waits for LDAP requests from the LDAP server before timing out. When an LDAP response is not received from the LDAP server

within the time specified, the request is retried again based on the max-request-timeouts parameter value. Valid values are 1 to 300 seconds. Default is 15.

XML Tag: timeLimit

```
ACMEPACKET(ldap-config)# timeout-limit 0
```

- max-request-timeouts—Enter the maximum number of times that the LDAP Server is sent LDAP requests before the ESD determines that the server is unreachable and terminates the TCP/TLS connection. When an LDAP response is not received within the time specified for the timeout-limit parameter value, the request is retried the number of times specified for this max-request-timeouts value. Valid values are 0 to 10. Default is 3.

  XML Tag: maxReqTimeouts

  ```
  ACMEPACKET(ldap-config)# max-request-timeouts 3
  ```
- tcp-keepalive—Specify whether or not the ESD keeps the TCP connection to the LPAD Server alive. Default is disabled. Valid values are:

  - enabled
  - disabled (default)

  XML Tag: tcpKeepalive

  ```
  ACMEPACKET(ldap-config)# tcp-keepalive enabled
  ```
- ldap-sec-type—Specify the LDAP security type to use when the ESD accesses the LDAP server. This parameter enables the use of LDAP over TLS (LDAPS). If you set a value for this parameter, you must also specify an ldap-tls-profile value. Default is none. Valid values are:

  - none (default) - No LDAP security type specified.
  - ldaps - Method of securing LDAP communication using an SSL tunnel. This is denoted in LDAP URLs. The default port for LDAP over SSL is 636.

  XML Tag: ldapSecType

  ```
  ACMEPACKET(ldap-config)# ldap-sec-type ldaps
  ```
- ldap-tls-profile—Enter the name of the Transport Layer Security (TLS) profile that the ESD uses when connecting to the LPAD Server. The ldap-sec-type must be set with an ldaps value for the LDAP configuration to use this profile. Valid values are alpha-numeric characters. Default is blank.

  XML Tag: ldapTLSProfile

  ```
  ACMEPACKET(ldap-config)# ldap-tls-profile ldap-tls
  ```
- ldap-transactions—Subelement to ldap-config. For more information on this element, see *Configuring ldap-transactions*.

  XML Tag: ldapTransaction

  ```
  ACMEPACKET(ldap-config)# ldap-transactions
  ACMEPACKET(ldap-transactions)#
  ```

XML Example for ldap-config

```
<ldapConfig name='ldapquery'
        state='enabled'
        ldapServers='172.44.0.20:636'
        realm='net172'
        authType='Simple'
        username='ENGLAB\Administrator'
        password='sips1234'
        ldapSearchBase='CN=Users,DC=englab,DC=acmepacket,DC=com'
        timeLimit='0'
        maxReqTimeouts='3'
        tcpKeepalive='enabled'
        ldapSecType='LDAPS'
        ldapTlsProfile='ldap-tls'
        lastModifiedBy='admin@console'
```

```
        lastModifiedDate='2012-06-28 20:25:13'
        objectId='102'>
        <key>ldapquery</key>
</ldapConfig>
```

Configuring ldap-transactions

You use the ldap-transactions object in the ACLI to configure the application transaction type for LDAP, determine route priority in the route list, and configure the LDAP configuration attributes. You configure this object for LDAP search queries in call routing.

To configure ldap-transactions:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session router-related objects.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type ldap-config and press Enter to access the LDAP configuration-related attributes.

   ```
   ACMEPACKET(session-router)# ldap-config
   ACMEPACKET(ldap-config)#
   ```

4. Type ldap-transactions and press Enter to access the LDAP transactions-related attributes.

   ```
   ACMEPACKET(ldap-config)# ldap-transactions
   ACMEPACKET(ldap-transactions)#
   ```

5. app-trans-type—Specify the application transaction type to use for LDAP. This value allows the ESD to add call routing updates to the Active Directory. Default is ad-call-routing. Valid values are:

   - ad-call-routing (default)

   XML Tag: type

   ```
   ACMEPACKET(ldap-transactions)# app-trans-type ad-call-routing
   ```

6. route-mode—Specify the route priority that the ESD uses in the route list. This parameter determines which routes are created, and the priority of those routes within the route list. Default is exact-match-only. Valid values are:

   - exact-match-only (default) - If there is an exact match between the dialed telephone number and an LDAP attribute value in the search response entry, a route is created corresponding to that LDAP attribute. If there is an exact match on multiple attributes, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route. For example, an enterprise that uses the same phone number for both Lync and IP-PBX phones, if the msRTCSIP-Line attribute is configured first, the corresponding next hop (Lync Server) would be used to create the first route in the route list.
   - attribute-order-only - The ordering of LDAP attributes in the LDAP configuration determines the priority for each route. So if the msRTCSIP-Line attribute is configured first, the corresponding next hop (Lync Server) would be used to create the first route in the route list. If there is a valid value present in the search response entry for a LDAP attribute, a route is created corresponding to that LDAP attribute.

     Note: The LDAP attribute must have a valid value in the response; a match is not necessary for that attribute. If an entry is returned in the search response, there must be a match on at least one other attribute. For example, the dialed telephone number could be +17813284392 (IP-PBX Phone#), and the msRTCSIP-Line in the response could be +17814307069 (Lync phone#). A route is created for the Lync phone#, even though the dialed telephone number is the PBX Phone#.
   - exact-match-first - If there is an exact match between the dialed telephone number and an LDAP attribute value in the search response entry, the corresponding route gets the highest priority in the route list. For the rest of the routes, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route. So if the msRTCSIP-Line attribute is configured first, the corresponding next hop (Lync Server) would be used to create the second highest priority route in the route list. If there is a valid value present in the search response entry for an LDAP attribute, a route is created corresponding to that LDAP attribute.

Note: The LDAP attribute must have a valid value in the response; a match is not necessary for that attribute. If an entry is returned in the search response, there must be a match on at least one other attribute. For example, the dialed telephone number could be +17813284392 (IP-PBX Phone#), and the msRTCSIP-Line in the response could be +17814307069 (Lync phone#). A route is created for the Lync phone#, even though the dialed telephone number is the PBX Phone#.

XML Tag: routeMode

```
ACMEPACKET(ldap-transactions)# route-mode exact-match-only
```

7. ldap-cfg-attributes—Subelement to ldap-config. For more information on this element, see *Configuring ldap-cfg-attributes*.

   XML Tag: ldapCfgAttribute

   ```
   ACMEPACKET(ldap-transactions)# ldap-cfg-attributes
   ACMEPACKET(ldap-cfg-attributes)#
   ```

   XML Example for ldap-transactions

   ```
   <ldapTransactions name='ldapquery'
           type='ad-call-routing'
           routeMode='exact-match-only'
   </ldapTransactions>
   ```

Configuring ldap-cfg-attributes

You use the ldap-cfg-attributes object in the ACLI to configure the Active Directory attribute name, next hop for routing SIP requests, the realm for the next hop, a regular expression pattern, and a format for the attribute value. You configure this object for LDAP search queries in the Active Directory.

To configure ldap-cfg-attributes:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session router-related objects.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type ldap-config and press Enter to access the LDAP configuration-related attributes.

   ```
   ACMEPACKET(session-router)# ldap-config
   ACMEPACKET(ldap-config)#
   ```

4. Type ldap-transactions and press Enter to access the LDAP transactions-related attributes.

   ```
   ACMEPACKET(ldap-config)# ldap-transactions
   ACMEPACKET(ldap-transactions)#
   ```

5. Type ldap-cfg-attributes and press Enter to access the LDAP configuration attributes.

   ```
   ACMEPACKET(ldap-transactions)# ldap-cfg-attributes
   ACMEPACKET(ldap-cfg-attributes)#
   ```

   attribute-name—Enter the Active Directory attribute name. Default is blank. Valid values are alpha-numeric characters. Some examples of Active Directory attribute names are:

   - ipPhone and msRTCSIP-Line for Lync phone number
   - telephoneNumber for IP PBX phone number
   - mobile for Mobile phone number

     XML Tag: name

     ```
     ACMEPACKET(ldap-cfg-attributes)# attribute-name msRTCSIP-Line
     ```

     next-hop—Enter the Active Directory's next hop when routing SIP requests. Default is blank. Valid values are alpha-numeric characters. Some examples of the Active Directory's next hop are:
   - SAG (Session Agent Group) name, specified by entering an sag: prefix

- SA (Sesson Agent) name
- IP Address

XML Tag: nextHop

```
ACMEPACKET(ldap-cfg-attributes)# next-hop sag:SA1
```

realm—Enter the name of the realm associated with the next hop. This value determines the network interface to which to route the SIP request. Valid values are alpha-numeric characters. Default is blank.

XML Tag: realm

```
ACMEPACKET(ldap-cfg-attributes)# realm net165
```

extraction-regex—Enter the regular expression pattern used to break down the string of digits in the phone number extracted from the request URI of the SIP request. The variables extracted from the phone number can be used in the attribute-value-format parameter. The default regex is "^\+?1?(\d{2})(\d{3})(\d{4})$". This value assumes that the phone number is a North American phone number specified in the E.164 format. It extracts three variables from the phone number:

- $1 is the area code
- $2 and $3 are the next 3 and 4 digits in the phone number

Valid values are alpha-numeric characters.

XML Tag: extractionRegex

```
ACMEPACKET(ldap-cfg-attributes)# extraction-regex ^\+?1?(\d{2})(\d{3})
(\d{4})$
```

attribute-value-format—Enter the format for the attribute value. These format values are extracted from the phone number using the extraction-regex parameter. The default parameter is "tel:+1$1$2$3". This value assumes that the phone number is a North American phone number specified in the E.164 format, and it recreates the phone number in E.164 format.

In addition to the E.164 format, Acme Packet's Active Directory uses other formats as well to store the phone numbers. You can customize the value specified for this parameter to enable successful queries for phone numbers in other formats.

Valid values are alpha-numeric characters.

XML Tag: valueFormat

```
ACMEPACKET(ldap-cfg-attributes)# attribute-value-format tel:+1$1$2$3
```

XML Example for ldap-cfg-attributes

```
<ldapCfgAttributes name='ldapquery'
        name='msRTCSIP-Line'
        nextHop='sag:SA1'
    realm='net1651'
    extractionRegex='^\+?1?(\d{2})(\d{3})(\d{4})$'
    attribute-value-format='tel:+1$1$2$3'
</ldapCfgAttributes>
```

Configuring policy-attributes

You use the policy-attributes object in the ACLI to configure the ldap: prefix with the name of the ldap-config. This allows the ESD to send LDAP queries to the Active Directory server(s) configured in the ldap-config element whenever there is a match for the corresponding local-policy.

Note: An ldap-config with the LDAP name specified for this value must be configured for the next hop. An LDAP next hop is supported only for SIP to SIP calls.

To configure policy-attributes for LDAP:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type local-policy and press Enter to access the local policy configuration-related attributes.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type policy-attributes and press Enter to access the policy attributes configuration-related attributes.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-attributes)#
```

next-hop—Enter the “ldap:” prefix along with the name of the ldap-config. An ldap-config with this name must be configured. An ldap next hop is supported only for SIP-to-SIP calls. Valid values are alpha-numeric characters. Default is blank.

XML Tag: nextHop

```
ACMEPACKET(ldap-cfg-attributes)# next-hop ldap:ldapquery
```

XML Example for policy-attributes for LDAP

```
<localPolicy description=''
   activateTime=''
   deactivateTime=''
   state='enabled'
   anonymousPriority='none'
   lastModifiedBy='admin@10.1.20.147'
   lastModifiedDate='2012-07-12 20:10:30'
   objectId='85'>
   <from addr='*'
      type='Hostname'
      addrPrefix=''/>
   <to addr='*'
      type='Hostname'
      addrPrefix=''/>
   <sourceRealm name='net192'/>
   <policyAttribute nextHop='ldap:ldapquery'
      destRealm='net172'
      action='none'
      isTermRoute='disabled'
      carrierName=''
      startTime='0000'
      endTime='2400'
      dow='U-S'
      cost='0'
      state='enabled'
      appProtocol='SIP'
      methods=''
      mediaProfiles=''
      lookup='single'
      nextKey=''
      eLocStrLkup='disabled'
      eLocStrMatch=''/>
   <key>view_realm_from_to/net192/Hostname/Hostname</key>
</localPolicy>
```

LDAP Error Messages

The ESD displays errors messages if the LDAP configuration objects are not properly configured. The following error messages for LDAP may display:

For all ldap-config objects:

- if an ldap-tls-profile is specified, and a tls-profile with that name has not been configured, the following error displays:

  ERROR: ldap-config [xyz] has reference to tls-profile [abc] which does not exist.
- if a realm has not been configured for an ldap-config, the following error displays:

  ERROR: ldap-config [xyz] has reference to realm [abc] which does not exist.

For all ldap-cfg-attributes:

- if a realm has not been configured for an ldap-config, the following error displays:

  ERROR: ldap-config [xyz] has reference to realm [abc] which does not exist.

For local policy-attributes:

- if the ldap-config object is configured corresponding to every ldap-config specified in the next-hop(s) in all policy-attribute subelements, and the
  next-hop value is not recognized, the following error displays:

  ERROR: local-policy-attribute [route; ldap:ldap-config-name] from local-policy [xyz] has reference to next-hop [ldap:ldap-config-name] which does not exist
- if the ldap-config object is not enabled, the following error displays:

  ERROR: local-policy-attribute [route; ldap:ldap-config-name] from local-policy [xyz] has reference to next-hop [ldap:ldap-config-name] which is not enabled

LDAP Show Commands

The ESD rovides specific LDAP statistics you can display using the show ldap command in the ACLI. The following is an example of the show ldap command output. These statistics include LDAP status information over Period and Lifetime monitoring spans, as well as information on active LDAP sessions. LDAP search query information displays for a Lifetime monitoring span only.

```
ACMEPACKET# show ldap
LDAP Status                 -- Period -- -------- Lifetime --------
                  Active    High   Total       Total  PerMax    High
Client Trans           0       0       0           0       0       0
Server Trans           0       0       0           0       0       0
Sockets                1       1       0           1       1       1
Connections            0       0       0           0       0       0

                          ---- Lifetime ----
                     Recent      Total  PerMax
Query                     0          0       0
Modify                    0          0       0
LDAP Requests             0          0       0
LDAP Errors               0          0       0
LDAP Rejects              0          0       0
LDAP Expires              0          0       0
LDAPD Errors              0          0       0
```

The following table describes each column in the above output.

Column Heading	Description
LDAP Status	
Client Trans	Total number of ESD LPAD transactions currently occurring and those that have occurred over the lifetime of the Active Directory.
Server Trans	Total number of LDAP Server transactions currently occurring and those that have occurred over the lifetime of the Active Directory.

Column Heading	Description
Sockets	Total number of active and past sockets established from the Active Directory on the ESD to the LPAD Server.
Connections	Total number of active and past connections established from the Active Directory on the ESD to the LPAD Server.
LDAP Search Queries	
Query	Total number of LDAP queries that occurred in the Active Directory on the ESD.
Modify	Total number of modified LDAP call routes in the Active Directory.
LDAP Requests	Total number of LDAP call route Requests in the Active Directory.
LDAP Errors	Total number of errors that occurred for LDAP call routes in the Active Directory.
LDAP Rejects	Total number of LDAP call routes that were rejected in the Active Directory.
LDAP Expires	Total number of times LDAP timed out or expired in the Active Directory.
LDAPD Errors	Total number of errors that occurred for the LDAP Daemon (LDAPD) in the Active Directory.

ENUM Lookup

Telephone Number Mapping (ENUM from TElephone NUmber Mapping) is a suite of protocols used to unify the telephone system with the Internet by using E.164 addresses with the Domain Name System (DNS). With ENUM, an E.164 number can be expressed as a Fully Qualified Domain Name (FQDN) in a specific Internet infrastructure domain defined for this purpose (e164.arpa). E.164 numbers are globally unique, language independent identifiers for resources on Public Switched Telecommunication Networks (PSTNs). ITU-T recommendation E.164 is the international public telecommunication telephony numbering plan.

How ENUM Works

ENUM uses DNS-based architecture and protocols for mapping a complete international telephone number (for example, +1 202 123 1234) to a series of Uniform Resource Identifiers (URIs).

The protocol itself is defined in the document E.164 number and DNS (RFC 3761) that provides facilities to resolve E.164 telephone numbers into other resources or services on the Internet. The syntax of Uniform Resource Identifiers (URIs) is defined in RFC 2396. ENUM uses Naming Authority Pointers (NAPTR) records defined in RFC 2915 in order to identify available ways or services for contacting a specific node identified through the E.164 number.

Translating the Telephone Number

A telephone number is translated into an Internet address using the following steps:

1. The number is first stored in the following format, +1-202-555-1234. 1 is the country code for the United States, Canada, and the seventeen other countries that make up the North American Numbering Plan (NANP). The + indicates that the number is a complete, international E.164 telephone number.
2. All characters are removed except for the digits. For example, 12025551234.
3. The order of the digits is reversed. For example, 43215552021. The telephone number is reversed because DNS reads addresses from right to left, from the most significant to the least significant character. Dots are placed between each digit. Example: 4.3.2.1.5.5.5.2.0.2.1. In DNS terms, each digit becomes a zone. Authority can be delegated to any point within the number.
4. A domain (for example, e164.arpa) is appended to the end of the numbers in order to create a FQDN. For example,4.3.2.1.5.5.5.2.0.2.1.e164.arpa.
5. The domain name is queried for the resource records that define URIs necessary to access SIP-based VoIP.

Once the authoritative name server for that domain name is found, ENUM retrieves relevant records and uses that data to complete the call or service. For example, the number 12025551234 returns sip:my.name@bigcompany.com.

About NAPTR Records

ENUM uses NAPTR records for URI resource records. NAPTR records are used to translate E.164 addresses to SIP addresses. An example of a NAPTR record is:

```
$ORIGIN 4.3.2.1.5.5.5.2.0.2.1.e164.arpa.
IN NAPTR 100 10 "u" "sip+E2U"  "!^.*$!sip:phoneme@example.net!"
```

This example specifies that if you want to use the "sip+E2U" service, you should use sip:phoneme@example.net as the address.

The regular expression can be used by a telephone company to easily assign addresses to all of its clients. For example, if your number is +15554242, your SIP address is sip:4242@555telco.example.net; if your number is +15551234, your SIP address is sip:1234@555telco.example.net.

About the Oracle Enterprise Session Border Controller ENUM Functionality

The ENUM functionality lets the Oracle Enterprise Session Border Controller make an ENUM query for a SIP request. The ENUM lookup capability lets the Oracle Enterprise Session Border Controller transform E.164 numbers to URIs during the process of routing (or redirecting) a call. During the routing of a SIP call, the Oracle Enterprise Session Border Controller uses a local policy attribute to determine if an ENUM query is required and if so which ENUM server(s) need to be queried. A successful ENUM query results in a URI that is used to continue routing or redirecting the call.

Configurable Lookup Length

You can configure a lookup length in the ENUM configuration that provides for more efficient caching of URI lookup results; in it, you can specify the length of the string for the DNS request starting from the most significant digit. This provides more flexibility for length matching, which is useful given the amount of wild card matching available in ENUM services. Specific ENUM groups might only be intended to provide NPANXX or wild card results.

UDP Datagram Support for DNS NAPTR Responses

The Oracle Enterprise Session Border Controller's default behavior is to conform to the DNS standard defined in RFC 1035 Domain Names: Implementation and Specification, which sets a maximum size for UDP responses of 512 bytes. This limitation means that responses larger than 512 bytes are truncated (set with the TC, or truncation, bit). In addition, this limitation protects network and system resources because using TCP consumes an undesirable amount of both.

However, you can configure support ENUM queries that manage larger UDP DNS responses as set out in RFC 2671, Extension Mechanisms for DNS (EDNS0), enabling your Oracle Enterprise Session Border Controller to manage responses beyond 512 bytes. According to RFC 2671, senders can advertise their capabilities using a new resource record (OPT pseudo-RR), which contains the UDP payload size the sender can receive. When you specify a maximum response size over 512 bytes, then the Oracle Enterprise Session Border Controller add the OPT pseudo-RR to the ENUM query—without which the ENUM server will truncate the response.

Custom ENUM Service Type Support

You can configure the ENUM service type that you want to use for an ENUM group. The Oracle Enterprise Session Border Controller has always supported E2U+sip and sip+E2U by default, and still does. With Release S-C6.1.0, however, you are also able to configure the service type to those supported in RFCs 2916 and 3721.

For example, you can now set the service type in the ENUM configuration to support E2U+sip and E2U +voicemsg:sip. When you configure customer ENUM service types on your system, however, you should note the following:

- New entries in the service-type parameter overwrite pre-existing values, including the default values.

- Because of the overwriting noted above, you must include the defaults (if you want them configured) when you are adding additional ENUM service type support. That is, you have to also type in E2U+sip and sip+E2U if you want them to be used in addition to the customized types you are setting.

ENUM Failover and Query Distribution

ENUM Query Distribution

The Oracle Enterprise Session Border Controller can intelligently distribute ENUM queries among all configured ENUM servers. By setting the enum config's query method parameter to round robin, the Oracle Enterprise Session Border Controller will cycle ENUM queries, sequentially, among all configured ENUM servers. For example, query 1 will be directed to server 1, query 2 will be directed to server 2, query 3 will be directed to server 3, and so on.

The default query method, hunt, directs all ENUM queries toward the first configured ENUM server. If the first server is unreachable, the Oracle Enterprise Session Border Controller directs all ENUM queries toward the next configured ENUM server, and so on.

Failover to New enum-config

When an enum-config's configured servers are unreachable via the network, i.e., no response is received on a query, the Oracle Enterprise Session Border Controller can failover to a defined ENUM config that contains different enum servers to query. This failover behavior works when all servers in an enum config are unreachable, rather than when the Oracle Enterprise Session Border Controller receives not-found type responses.

The Oracle Enterprise Session Border Controller queries each ENUM server once before trying the next configured server, and then ultimately trying the servers listed in the failover-to enum config. If the failover-to servers also are unreachable, the Oracle Enterprise Session Border Controller fails the call; the failover-to behavior does not recurse among enum-configs, it only checks the first, linked enum-config.

ENUM Server Operation States

After 5 consecutive failed attempts, an ENUM server is considered Out of Service (OOS). All subsequent queries which would be directed to the OOS servers are immediately directed to the first non-OOS server. ENUM servers return to in-service after 600 seconds. If all configured ENUM servers are OOS, the Oracle Enterprise Session Border Controller fails the call.

After the first failed attempt to reach an ENUM server, it is placed in a Time Out state, which it stays in for 30 seconds. Within this 30 seconds it will not be contacted when an ENUM query is made. After the 30 seconds pass, the ENUM server goes back to an in-service state.

Server Availability Monitoring

The Oracle Enterprise Session Border Controller can probe an ENUM server's health by sending it a standard ENUM NAPTR query and receiving a valid answer. The query is for the phone number defined in the health query number parameter, which should be one that the ENUM servers can positively resolve. As long as the query succeeds, that ENUM server maintains its in-service state and is available for ENUM queries. Any lack of response, whether network based (time-outs), or application based (DNS error or not found response) is considered a query failure and the server is set to OOS and unavailable for ENUM queries.

The Oracle Enterprise Session Border Controller continuously checks the health of all configured ENUM servers to determine their current state and monitor for failed servers' return to service. All servers are checked for availability at the health query interval parameter, as defined in seconds.

Note: When ENUM server availability monitoring is enabled, ENUM servers can only exist in an in-service or out-of-service states; Without the health query interval defined, server availability monitoring is disabled, and ENUM servers exist in three service states.

ENUM Server IP Address and Port

You can configure an IP address and port for each enum server listed in the enum-servers parameter. IP address and port are specified in XXX.XXX.XXX.XXX:YYYY format with a port value range of 1024-65535. If the port number is not specified, 53 is assumed.

The Oracle Enterprise Session Border Controller supports IPv6 ENUM configurations in IPv6 realms. The enumservers parameter in the enum-config configuration parameter can be configured IPv6 addresses in addition to IPv4 addresses. When IPv6 Addresses are used, the realm configured in the realm-id parameter must be an IPv6 realm.

Caching ENUM Responses

As DNS responses often lead to further DNS queries, a DNS server can send additional multiple records in a response to attempt to anticipate the need for additional queries. The Oracle Enterprise Session Border Controller can locally cache additional NAPRT, SRV, and A records returned from an ENUM query to eliminate the need for unnecessary external DNS requests by enabling the cache addl records parameter. These cached records can then be accessed by internal ENUM and DNS agents.

The unprompted NAPTR, SRV, or A record returned to the Oracle Enterprise Session Border Controller must include complete information to resolve a call to be added to the local DNS/ENUM cache, otherwise the Oracle Enterprise Session Border Controller will preform an external query to fine the address it is looking to resolve.

Cached entries are per ENUM config. That means if one ENUM config has a number of cached entries, and an ENUM request is directed through a different ENUM config, the second configuration is not privy to what the first configuration has cached.

The Oracle Enterprise Session Border Controller uses the shorter lifetime of the DNS response's TTL or the server dns attribute's transaction-timeout to determine when to purge a DNS record from the local cache.

Source URI Information in ENUM Requests

ENUM queries can be configured to include the source URI which caused the ENUM request by enabling the include source info parameter. The Oracle Enterprise Session Border Controller can add the P-Asserted-ID URI (only if not in an INVITE) or the From URI into an OPT-RR Additional Record to be sent to the ENUM server. It can be useful to specify the originating SIP or TEL URI from a SIP request which triggered the ENUM query, so the ENUM server can provide a customized response based on the caller.

This feature implements the functionality described in the Internet Draft, DNS Extension for ENUM Source-URI, draft-kaplan-enum-source-uri-00.

When a P-Asserted-ID is blocked or removed before the ENUM query is made, the Oracle Enterprise Session Border Controller only sends the URI in the From header.

Note that to support this feature, according to the Internet draft, ENUM clients must support 1220 bytes in UDP responses. Therefore, if this feature is enabled, and the max response size parameter is not set i.e., with a 512 byte default, the Oracle Enterprise Session Border Controller will set the size to 1200 on the OPT-RR records sent.

Operation Modes

There are four modes of ENUM operation that are selected on a global basis:

- stateless proxy
- transaction stateful proxy
- session stateful proxy
- B2BUA with or without media

Stateless Proxy Mode

The stateless proxy mode is the most basic form of SIP operation. The stateless proxy mode:

- Has the least number of messages per call. No record route header is added and there are no 100 Trying or BYEs.

- Does not keep transaction state (timers and retransmission). There are no session counters and no session stop time. No session stop time means no RADIUS STOP records.
- Has no limits on session state.
- Can restrict functionality by specification. This can mean no media management, limited potential for RADIUS accounting, and no CALEA (no Release/BYE messages for CDC).
- Acts primarily as a routing device, with local policy routing and ENUM routing.

Transaction Stateful Proxy

In the transaction stateful proxy mode:

- Adds state to the proxy (not dialogs).
- Has lower number of messages per call. No Record Route header added and no BYES.
- Keeps transaction state (timers and retransmissions.
- Enforces session restrictions (32k) because of state management. These restrictions can be increased.
- Can restrict functionality by specification. This can mean no media management, limited potential for RADIUS accounting, and no CALEA (no Release/BYE message for CDC).
- Acts as routing device with transaction timers, with local policy routing and ENUM routing.
- Can off-load some transactions across unreliable links.

Session Stateful Proxy

The session stateful proxy mode:

- Maintains dialog state as a proxy.
- Includes BYES (though cannot be inserted)
- Keeps transaction state (timers and retransmission)
- Provides per-session information such as session counters per session agent, RADIUS STOP accounting record generation, CALEA CDC generation.
- Enforces session restrictions (32k) because of state management.
- Does not provide media management. There is no CALEA CCC.
- Routes full sessions with transaction timers with local policy routing and ENUM routing.

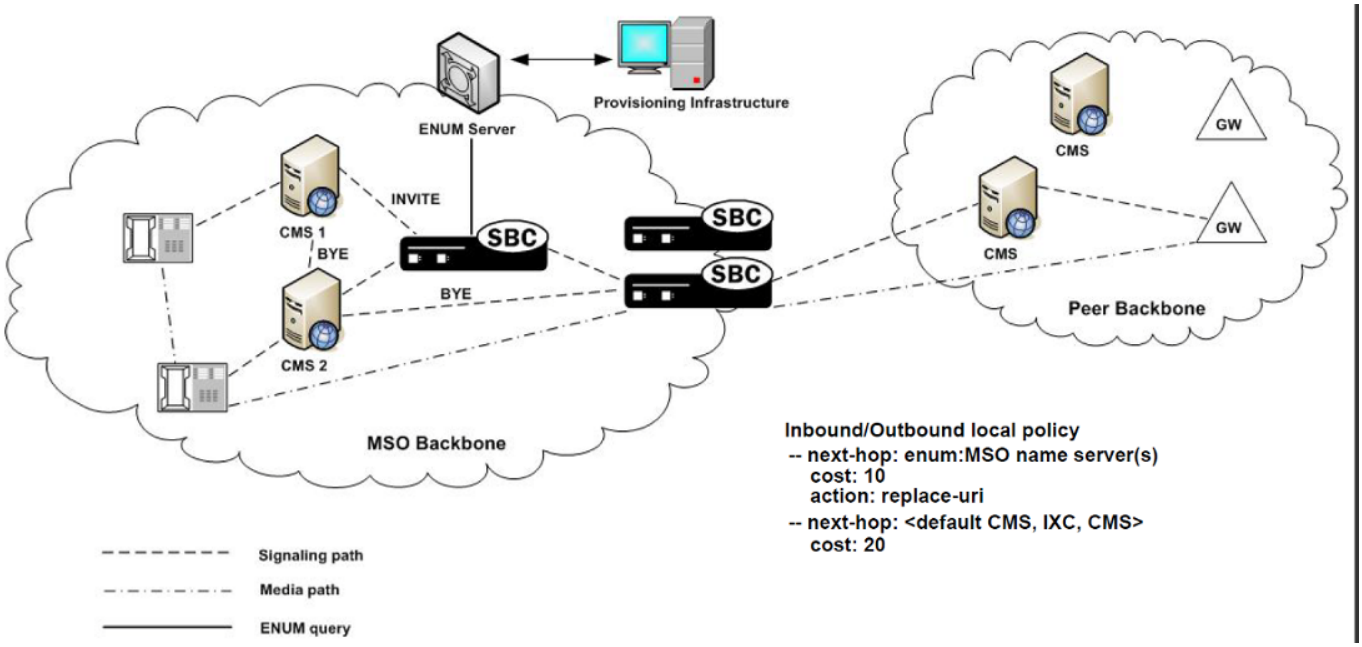
B2BUA

The B2BUA mode:

- Acts as UAS and UAC within call flow.
- Includes BYES (can be inserted).
- Keeps transaction state (timers and retransmissions)
- Provides per-session information such as session counters per session agent, RADIUS STOP accounting record generation, CALEA CDC generation.
- Enforces session restrictions (32k) because of state management.
- Can provide media management, including media routing through a single IP address with topology masking, CALEA CCC, media watchdogs for state management.
- Routes full sessions with topology masking. Includes rewriting Via, Route, Contact headers, full NATing with SIP NAT or header manipulation, direct bridging, local policy routing, and ENUM routing.

Example ENUM Stateless Proxy

The following diagram shows the Oracle Enterprise Session Border Controller using ENUM to query a local subscriber database. The Oracle Enterprise Session Border Controller serves as the inbound and outbound routing hub and performs media management. Calls are routed throughout the MSO network using ENUM lookup results.



ENUM Configuration

This section shows you how to configure ENUM on your Oracle Enterprise Session Border Controller.

To configure ENUM:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type enum-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```

4. name—Enter a string that uniquely identifies this ENUM configuration. You use this name in other areas of the Oracle Enterprise Session Border Controller configuration to refer to this ENUM configuration. For example, in the local policy attributes.
5. top-level-domain—Enter the domain extension to be used when querying the ENUM servers for this configuration. For example, e164.arpa. The query name is a concatenation of the number and the domain.

   For example the number is +17813334444 and the domain is e164.arpa, the query name would be 4.4.4.4.3.3.3.1.8.7.1.e164.arpa.com.
6. realm-id—Enter the realm where the ENUM servers can be reached. The realm ID is used to determine on which network interface to issue the ENUM query.
7. enum-servers—Enter the list of ENUM servers (an ENUM server and corresponding redundant servers) to be queried. Separate each server address with a space and enclose list within parentheses.

   The first server on this list is the first one to be queried. If the query times out (including retransmissions) without getting a response, the next server on the list is queried and so on.
8. service-type—Enter the ENUM service types you want supported in this ENUM configuration. Possible entries are E2U+sip and sip+E2U (the default), and the types outlines in RFCs 2916 and 3721.

   This parameter defaults to the following service types: E2U+sip and sip+E2U.

You can enter multiple services types in the same entry, as in this example:

```
ACMEPACKET(enum-config)# service-type E2U+sip,sip+E2U,E2U+voicemsg
```

9. query-method—Set the strategy the Oracle Enterprise Session Border Controller uses to contact ENUM servers. Valid values are:

   - hunt—Directs all ENUM queries toward the first configured ENUM server. If the first server is unreachable, the Oracle Enterprise Session Border Controller directs all ENUM queries toward the next configured ENUM server, and so on.
   - round-robin—Cycles all ENUM queries, sequentially, among all configured in-service ENUM servers. Query 1 will be directed to server 1, query 2 will be directed to server 2, query 3 will be directed to server 3.

10. timeout—Enter the total time in seconds that should elapse before a query sent to a server (and its retransmissions) will timeout. If the first query times out, the next server is queried and the same timeout is applied. This process continues until all the servers in the list have timed out or until one of the servers responds.

    The retransmission of ENUM queries is controlled by three timers. These timers are derived from this timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the Oracle Enterprise Session Border Controller should retransmit 3 times before timing out to give the server a chance to respond. The valid values are:

    - Init-timer—Is the initial retransmission interval. If a response to a query is not received within this interval, the query is retransmitted. To safeguard from performance degradation, the minimum value allowed for this timer is 250 milliseconds.
    - Max-timer—Is the maximum retransmission interval. The interval is doubled after every retransmission. If the resulting retransmission interval is greater than the value of max-timer, it is set to the max-timer value.
    - Expire-timer—Is the query expiration timer. If a response is not received for a query and its retransmissions within this interval, the server will be considered non-responsive and the next server in the list will be tried.

      The following examples show different timeout values and the corresponding timers derived from them.

      timeout >= 3 seconds

      ```
      Init-timer = Timeout/11
      Max-Timer = 4 * Init-timer
      Expire-Timer = Timeout
      ```

      timeout = 1 second

      ```
      Init-Timer = 250 ms
      Max-Timer = 250 ms
      Expire-Timer = 1 sec
      ```

      timeout = 2 seconds

      ```
      Init-Timer = 250 ms
      Max-Timer = 650 ms
      Expire-Timer = 2sec
      ```

11. cache-inactivity-timer—Enter the time interval in seconds after which you want cache entries created by ENUM requests deleted, if inactive for this interval. If the cache entry gets a hit, the timer restarts and the algorithm is continued until the cache entry reaches its actual time to live.

    Setting this value to zero disables caching. For optimal performance, set this to one hour. Rarely used cache entries are purged and frequently used entries are retained. The default value is 3600. The valid range is:

    - Minimum—0
    - Maximum—999999999

12. lookup-length—Specify the length of the ENUM query, starting from the most significant digit. The default is 0. The valid range is:

    - Minimum—1
    - Maximum—255

13. max-response-size—Enter the maximum size in bytes for UDP datagrams in DNS NAPTR responses. This parameter takes values from 512 (default) to 65535. Although the maximum value you can set is 65535, Oracle recommends configuring values that do not exceed 4096 bytes.
14. health-query-number—Set this parameter to a standard ENUM NAPTR query that will consistently return a positive response from the ENUM server.
15. health-query-interval—Set this parameter to the number of seconds to perpetually probe ENUM servers for health.
16. failover-to—Set this parameter to the name of another ENUM-config which to failover to under appropriate conditions.
17. cache-addl-records—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to add additional records received in an ENUM query to the local DNS cache.
18. include-source-info—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to send source URI information to the ENUM server with any ENUM queries.
19. Save your work.

Example

The following example shows an ENUM configuration called enumconfig.

```
enum-config
        name                            enumconfig
        top-level-domain
        realm-id                        public
        enum-servers                    10.10.10.10:3456
                                        10.10.10.11
        service-type                    E2U+sip,sip+E2U
        query-method                    hunt
        timeout                         11
        cacheInactivityTimer            3600
        max-response-size               512
        health-query-number             +17813245678
        health-query-interval           0
        failover-to                     enumconfig2
        cache-addl-records              enabled
        include-source-info             disabled
```

Configuring the Local Policy Attribute

You can specify that an ENUM query needs to be done for the routing of SIP calls. You do so by configuring the local policy's next-hop attribute with the name of a specific ENUM configuration, prefixed with the enum: tag. For example: enum:test

You can configure multiple next-hops with different ENUM servers or server groups (possibly with different top-level-domains). If the first ENUM server group you enter as the next hop is not available, one of the others can be used.

Note: A new parameter called action has replaced the policy attribute's replace-uri parameter available prior to build 211p19.

To configure local policy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type local-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# local-policy
   ACMEPACKET(local-policy)#
   ```

4. next-hop—Enter the name of the ENUM configuration with the prefix enum:. For example, enum:test.
5. action—Set to redirect if you want to send a REDIRECT message back to the calling party with the information returned by ENUM in the Contact. The calling party then needs to send a REDIRECT using that information. The default value is none. Valid values are:
   - none—No specific actions requested.
   - replace-uri—To replace the next Request-URI with the next hop.
   - redirect—To send a redirect response with this next hop as contact.
6. Save and activate your configuration.

Local Policy Example

The following example shows one local policy with the next-hop configured to use enum:test and a second with the next-hope configured to use enum:test_alternate.

```
local-policy
     from-address            *
     to-address              *
     source-realm            public
     activate-time           N/A
     deactivate-time         N/A
     state                   enabled
     last-modified-date      2006-03-09 09:18:43
     policy-attribute
       next-hop                enum:test
       realm                   public
       action                  none
       terminate-recursion     disabled
       carrier
       start-time              0000
       end-time                2400
       days-of-week            U-S
       cost                    1
       app-protocol            SIP
       state                   enabled
       media-profiles
     policy-attribute
       next-hop                enum:test_alternate
       realm                   public
       action                  none
       terminate-recursion     disabled
       carrier
       start-time              0000
       end-time                2400
       days-of-week            U-S
       cost                    2
       app-protocol            SIP
       state                   enabled
```

CNAM Subtype Support for ENUM Queries

CNAM, calling name, data is a string up to 15 ASCII characters of information associated with a specific calling party name. The *Internet-draft, draft-ietf-enum-cnam-08.txt*, registers the Enumservice 'pstndata' and subtype 'cnam' using the URI scheme 'pstndata:' to specify the return of CNAM data in ENUM responses. The Oracle Enterprise Session Border Controller recognizes CNAM data returned via this mechanism. CNAM data is then inserted into the display name of the From: header in the original Request. If a P-Asserted-ID header is present in the original request, the CNAM data is inserted there as well.

CNAM data is identified by an ENUM response with service-type: E2U+pstndata:cnam

CNAM support is configured in the sip profile configuration element, which can then be applied to either a session agent, realm, or SIP interface.

The Oracle Enterprise Session Border Controller can preform CNAM queries on the signaling message's ingress or egress from the system by setting the cnam lookup direction parameter to either ingress or egress. If the CNAM lookup direction parameters are configured on both the ingress and egress sides of a call, the Oracle Enterprise Session Border Controller will only preform the lookup on the ingress side of the call.

CNAM Unavailable Response

A CNAM response can include a Calling Name Privacy Indicator parameter ('unavailable=p') or Calling Name Status Indicator parameter ('unavailable=u') in responses. The Oracle Enterprise Session Border Controller can insert a custom reason string into the SIP message's From and P-Asserted-ID header in the original requires.

Configuring the cnam unavailable ptype parameter inserts the specified text into the From and P-Asserted-ID headers when a CNAM response contains the unavailable=p parameter.

Configuring the cnam unavailable utype parameter inserts the specified text into the From and P-Asserted-ID headers when a CNAM response contains the unavailable=u parameter.

SIP Profile Inheritance

CNAM features, via the SIP Profile configuration element can be applied to session agents, realms, and SIP interfaces. The more generalized object inherits the more specific object's values. For example, if CNAM support via a SIP profile is configured on a session agent, the expected processing will override any SIP profile configuration on the downstream realm or SIP interface. Likewise, if CNAM support is unconfigured on the receiving session agent, but configured in the realm, CNAM configuration on the SIP interface will be ignored.

CNAM Subtype Support Configuration

To enable the Oracle Enterprise Session Border Controller to preform CNAM subtype ENUM queries, you must configure a SIP profile with an enum-config object (that points to valid ENUM servers). The referenced enum-config configuration element lists the servers to contact for CNAM type queries (and other general ENUM server interaction parameters).

To configure CNAM subtype support:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the signaling-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type sip-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-profile
   ACMEPACKET(sip-profile)#
   ```

4. name—Enter a string that uniquely identifies this SIP profile configuration. You use this name in other areas of the Oracle Enterprise Session Border Controller configuration to refer to this SIP profile in session agents, realms, or SIP interfaces.
5. cnam-lookup-server—Set this parameter to the name of an ENUM-config to that will query ENUM servers for CNAM data.
6. cnam-lookup-dir—Set this parameter to ingress or egress to identify where the Oracle Enterprise Session Border Controller performs a CNAM lookup with respect to where the call traverses the system. The default value is egress.
7. cnam-unavailable-ptype—Set this parameter to a string, no more than 15 characters, to indicate that the unavailable=p parameter was returned in a CNAM response.

8. cnam-unavailable-utype—Set this parameter to a string, no more than 15 characters, to indicate that the unavailable=u parameter was returned in a CNAM response.
9. Save your work.

Direct Inward Dial (DID)-Range-Based Local Routing Table (LRT)

The Oracle Enterprise Session Border Controllersupports LRT, an XML document that contains either E164 telephone numbers or strings-to-SIP-URI mappings. An iLRT is is configured and transferred from the development environment to the ESD /code/lrt directory. After installation and configuration, the LRT is available for SIP Request routing.

The information in the following sections is specific to contiguous ranges of Direct Inward Dial (DID) telephone numbers. Users of this LRT type should be acquainted with XML, and familiarize themselves with the more generic LRT descriptions supplied by the *SC6.4.0 ACLI Configuration Guide*.

Creating a DID-Range-Based LRT File

A DID-Range-Based LRT file is a well-formed XML document with a <localRoutes/> root element.

The following attribute is found within the <localRoutes/> element.

type — This attribute is required for a DID-range-based LRT; set this attribute's value to range.

<localRoutes/> can contain any number of child <route/> elements.

Each <route/> element contains:

- a required <user/> element that (1) defines the LRT type, and (2) in the case of a DID-range-based LRT, specifies a contiguous range of DID telephone numbers

  The following attributes are found within the <user/> element:

  type — This required attribute can be assigned one of three enumerated values (E164, string, or range); for a DID-range-based LRT, you must use the range value.

  rangeStart — This required attribute specifies the start value for the DID range.

  rangeEnd— This required attribute specifies the end value for the DID range.

  rangePrefix — This optional attribute specifies the common prefix for the range bracketed by the rangeStart and rangeEnd attributes.
- a required <next/> element that uses regular expression syntax to specify the routing next hop
- an optional <description/> element that provides information relevant to the range of DID addresses

When crafting your LRT file, keep the following rules in mind.

1. Set the type attribute of the <localRoutes/> root element to range.
2. Set the type attribute of all <user/> elements to range; <user/> elements of types other than range are invalid.
3. The start and end values of a range, must be valid E164 numbers.
4. The start and end values of a range, must contain the same number of digits.
5. The value of the rangeStart attribute must be less than, or equal to, the value of the rangeEnd attribute.
6. Ranges must not overlap. For example, the following ranges overlap.

   1000 - 1050

   1025 - 9999

   These ranges do not overlap.

   5510 -5519 (defines a contiguous range from 5510 through 5519)

55100 - 55199 (defines a contiguous range from 55510 through 55519)

7. After completing the LRT file, you must use FTP or SFTP to install the file in the ESD /code/lrt directory.

   The following annotated XML sample provides a template to assist users in crafting their own LRT file.

```
<?xml version=”1.0? encoding=”US-ASCII” standalone=”yes”>

<!-- A Local Routing Table (LRT) file based on DID ranges -->

<localRoutes type=”range”>  <!-- set to range for DID LRT -->

<!-- At least one <route/> element is required -->

<!-- A one number range -->
     <route>
<!-- <user/> element is required -->
          <user type=”range” rangeEnd=”5558888” rangeStart=”5558888”></user>
<!-- <description/> element is optional -->
          <description>CampusSecurity</description>
<!-- <next/> element is required -->
<!-- the next hop for CampusSecurity expressed as a regex -->
          <next type=”regex”>!^.*$!sip:1@public-range!</next>
     </route>

<!-- A multi-number range matches 149 through 155 -->
     <route>
          <user type=”range” rangeEnd=”155” rangeStart=”149”></user>
          <description>Quad 1</description>
<!-- the next hop for 149-to-155 expressed as a regex -->
          <next type=”regex”>!^.*$!sip:1@public-range!</next>
     </route>

<!-- A multi-number range with a prefix matches a1b149-to-a1b155 -->
     <route>
          <user type=”range” rangePrefix=”a1b” rangeEnd=”155”
           rangeStart=”149”></user>
          <description>Quad 1</description>
<!-- the next hop for aib149-to-aib155 expressed as a regex -->
          <next type=”regex”>!^.*$!sip:1@public-range!</next>
     </route>
</localRoutes>
```

Configuring a DID-Range-Based LRT

The following procedures provide information about configuring the LRT location and enabling LRT.

Specifying the LRT Location

After moving the DID-range-based LRT to the /code/lrt directory on the ESD, use the following procedure to specify the file’s location, and the lookup method.

1. Move to local-routing-config mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)# local-routing-config
ACMEPACKET(local-routing-config)#
```

2. Provide an alias for the LRT file. Later, you will use this alias to assign the LRT to the local policy attributes.

```
ACMEPACKET(local-routing-config)# name WestCampus
ACMEPACKET(local-routing-config)#
```

3. Provide the name of the file that contains the XML-formatted LTR. This file must currently exist within the /code/lrt directory.

```
ACMEPACKET(local-routing-config)# file-name didLRT.xml.gz
ACMEPACKET(local-routing-config)#
```

4. Specify the look-up type. For DID-based LRTs, enable string-lookup to ensure that all ranges, including those with an alphabetic prefix (for example, test123), are properly evaluated.

```
ACMEPACKET(local-routing-config)# string-lookup enabled
ACMEPACKET(local-routing-config)#
```

5. Because the prefix-length (if any) is specified within the XML file, ensure that the prefix-length attribute is set to its default value, 0.

```
ACMEPACKET(local-routing-config)# prefix-length 0
ACMEPACKET(local-routing-config)#
```

6. Retain default values for other parameters.
7. Use done, exit, and verify-config to complete this configuration.

Enabling LRT Usage

You enable LRT usage by assigning it to the local policy attributes.

Note: When enabling usage of a DID-range-based LRT, you need not specify a match-mode, as required for E164- or string-based LRTs. The match-mode parameter is ignored for DID LRTs.

1. Move to local-policy-attributes configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

2. Enable DID-based LRT usage by including the previously configured LRT alias in the local policy attributes list.

```
ACMEPACKET(local-routing-attributes)# next-hop lrt:WestCampus
ACMEPACKET(local-routing-attributes)#
```

3. To ensure string type lookups, verify that the eloc-str-lkup and eloc-str-match parameters are both enabled.

```
ACMEPACKET(local-routing-attributes)# eloc-str-lkup enabled
ACMEPACKET(local-routing-attributes)# eloc-str-match enabled
ACMEPACKET(local-routing-attributes)#
```

4. Use done, exit, and verify-config to complete this configuration.

Managing LRT using the Show LRT Command

Existing ACLI show commands (show lrt and show lrt route-entry) commands have been enhanced to support DID ranges. A new command (show lrt route-table) displays the contents of a DID-range-based LRT.

show lrt now provides a count of valid and invalid route ranges as shown below.

```
ESD01# show lrt
14:43:03-64183
Name: lroute
Local Route Statistics       ---- Lifetime ----
                        Recent      Total     PerMax
Queries                      0          0          0
Result - Success             0          0          0
Result - Not found           0          0          0

Valid Route Entries:        19
Invalid Route Entries:       0
Valid Route Ranges          15                     // New to Version E-
C[xz]6.4.0
Invalid Route Ranges         3                     // New to Version E-
C[xz]6.4.0
```

show lrt route-entry displays matching DID ranges as follows.

```
ESD01# show lrt route-entry lroute 323
UserName <320-329>
            Entry Type = range
            NextHop = !^.*$!sip:3@public-range!
            NextHop Type = regexp
            Description = Test DID range
```

The new show lrt route-table displays a DID-range-based table as follows.

```
ESD01# show lrt route-table la 2
UserName <320-329>
        Entry Type = range
        NextHop = !^.*$!sip:3@public-range!
        NextHop Type = regexp
        Description = Test DID range

UserName <3123 - 3125>
        Entry Type = range
        NextHop = !^.*$!sip:1@public-range!
        NextHop Type = regexp

Displaying 2 of 13 routes. Continue [y/n]?
```

LRT Entry Matching

When searching an LRT for a matching route, the Oracle Enterprise Session Border Controller can be configured with one of three match modes with the match mode parameter in the local routing config. These modes are:

- exact—When searching the applicable LRT, the search and table keys must be an exact match.
- best—The longest matching table key in the LRT is the chosen match.
- all—The all mode makes partial matches where the table's key value is a prefix of the lookup key. For example, a lookup in the following table with a key of 123456 returns entries 1, 3, and 4. The 'all' mode incurs a performance penalty because it performs multiple searches of the table with continually shortened lookup keys to find all matching entries. This mode also returns any exact matches too.

Entry#	Key	Result
1	1	<sip:\0@host1.example.com>
2	122	<sip:\0@host22.example.com>
3	123	<sip:\0@host3.example.com>
4	1234	<sip:\0@host4.example.com>
5	1234567	<sip:\0@host7.example.com>
6	1235	<sip:\0@host5.example.com>

LRT Entry Matching Configuration

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type local-routing-config and press Enter.

   ```
   ACMEPACKET(session-router)# local-routing-config
   ACMEPACKET(local-routing-config)#
   ```

4. match-mode—Set this parameter to either best, all, or leave it as exact which is the default. This indicates to the Oracle Enterprise Session Border Controller how to determine LRT lookup matches.

5. Save your work using the done command.

LRT String Lookup

The Oracle Enterprise Session Border Controller can search an LRT for either E.164 or string table keys. This selection is on a global basis. When the string-lookup parameter is disabled (default) in the local routing configuration, all lookups with be E.164 type, except when:

- If eloc-str-lkup is enabled in a matching local policy's policy-attribute, E-CSCF procedures are applied and the resulting lookup type is 'string'.
- The Oracle Enterprise Session Border Controller also performs string lookups exclusively when a compound lookup key is specified.

When the lookup type is 'E.164', the lookup is skipped if the lookup key is not a valid telephone number (i.e. it must contain only digits).

LRT String Lookup Configuration

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-routing-config and press Enter.

```
ACMEPACKET(session-router)# local-routing-config
ACMEPACKET(local-routing-config)#
```

4. string-lookup—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to perform LRT lookups on table keys of a string data type. Leave this parameter to its default as disabled to continue using E.164 type lookups.
5. Save your work using the done command.

Directed Egress Realm from LRT ENUM

A message can be sent into a specific egress realm identified in an ENUM query or LRT lookup. The egress realm is noted by a configurable parameter in the result URI. The Oracle Enterprise Session Border Controller is configured with the name of this parameter, that indicates an egress realm name, and looks for it in the returned URI.

To configure the parameter name, the egress-realm-param option is added to the sip config and/or the h323 config using the following format:

```
egress-realm-param=<name>
```

Where <name> is the parameter name to extract the egress realm name from.

When the egress realm param is defined, the ENUM and LRT results will always be checked for the presence of the URI parameter. The sip config options will apply for received SIP requests. The h323 config option will apply for received H.323 messages.

For example, if egress-realm-param=egress is added to the sip config, a matching entry in the LRT that specifies the egress realm core will look like this:

```
<route>
<user type="E164">+17815551212</user>
<next type="regex">!^.*$!sip:\0@core.example.com;egress=core!</next>
</route>
```

If the URI does not contain the parameter or the parameter identifies a realm that is not configured on the system, the egress realm that is normally applicable (from local policy, SIP-NAT, or session-agent data) will be used.

Directed Egress Realm Configuration

To add an egress parameter to look for in a sip-config:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. If you are adding this feature to a pre-existing SIP configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. egress-realm-param—Configure this option with the parameter to parse for in a returned ENUM or LRT result: For example

   - options egress-realm-param=egress

     In order to append the new option to the sip-config's options list, you must prepend the new option with a plus sign. For example:

     ```
     ACMEPACKET(sip-config)# options +egress-realm-param=egress
     ```

5. Save your work using the ACLI done command.

Note: The egress-realm-param option can be configured similarly in the h323-config.

SIP Embedded Route Header

The Oracle Enterprise Session Border Controller examines the ENUM and LRT lookup result for embedded Route headers. In the LRT or as returned in an ENUM query a URI including an embedded route header would look like:

```
<sip:user@example.com?Route=%3Csip:host.example.com;lr%3E>
```

Using embedded Route headers is the Oracle Enterprise Session Border Controller's default behavior. This can be overridden by adding the sip-config option use-embedded-route.

When the ENUM or LRT result becomes the top Route header, any embedded Route headers extracted are inserted just after that top Route (which will always be a loose route and include the "lr" URI parameter). In this case, the request will be sent to the top Route.

When the ENUM or LRT results become the Request-URI, any embedded Route headers extracted from the result are inserted before any Route headers already in the outgoing request. After that, if there are any Route headers in the outgoing request and the top Route header has an "lr" URI parameter, the request is sent to the top Route header. Otherwise, the request is sent to the Request URI.

SIP Embedded Route Header Configuration

To set the Oracle Enterprise Session Border Controller's default behavior of using embedded route headers from ENUM queries or LRT lookups:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. If you are adding this feature to a pre-existing SIP configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. use-embedded-route—Configure this as an option with one of the following arguments:

   - all = use embedded routes from both ENUM and LRT results (default)
   - none = do not use embedded routes
   - enum = use embedded routes from ENUM results only
   - lrt = use embedded routes from LRT results only

   In order to append the new option to the sip-config's options list, you must prepend the new option with a plus sign. For example:

   Set the options parameter by typing options, a Space, the option name use-embedded-route, and then press Enter.

```
ACMEPACKET(sip-config)# options +use-embedded-route=none
```

5. Save your work using the ACLI done command.

LRT Lookup Key Creation

This section describes the Oracle Enterprise Session Border Controller's LRT lookup key creation capability.

Arbitrary LRT Lookup Key

In addtion to the standard From, To, and and P-Asserted-Identity header fields the Oracle Enterprise Session Border Controller can now use the values from any arbitrary SIP header as an LRT or ENUM lookup key. This is preformed by prepending a dollar sign $ by the header name whose value's userinfo portion of the URI will be used as the lookup key. For example, key=$Refer-To would use the userinfo portion of the URI in the Refer-To header of the request as the lookup key.

An ampersand & followed by a header name will use the whole value of the header as the lookup key. For example, key=&X-Route-Key would use the whole value of the X-Route-Key as the lookup key. As a shortcut, an ampersand is not required for a "hidden" header. For example, "key=@LRT-Key" would use the value of the @LRT-Key header as the lookup key.

Hidden Headers for HMR and LRT lookup

When an LRT lookup key is more complex than just the URI's userinfo or a Tel-URI, HMR can be used to extract the data and build a special header.

By using a header name that begins with the at-sign "@" (e.g. @lrt-key), the header can be hidden and not included in outgoing SIP message, thus eliminating the need for an extra HMR rule to remove it.

Since '@' is not a valid character in a header name as defined by RFC 3261, there is no possibility of a collision between a header name defined in the future and a hidden header name beginning with @.

Compound Key LRT Lookup

LRT lookup keys can be combinations of more than one key value. For example, "key=$FROM,$TO" would construct a compound key with the userinfo of the From URI followed by a comma flowed by the userinfo of the To URI.

If the request message contained:

```
From: <sip:1234@example.com>
To: <sip:5678@example.com>
```

The compound key to match this From/To pair is "1234,5678".

In the table lookup, the compound key is a single key value and there is no special treatment of the comma in key matching. The comma is simply an ordinary additional character that is matched like any letter or digit (i.e. the comma must appear in the LRT entry's "type" element data). For example, if the table were configured for "best" match-mode, the lookup key "1234,5678" would match a table entry of "1234,567", but it would not match a table entry of "123,5678".

Retargeting LRT ENUM-based Requests

Request re-targeting is when a target or a request as indicated in the Request-URI, is replaced with a new URI.

The happens most commonly when the "home" proxy of the target user replaces the Request-URI with the registered contact of that user. For example, the original request is targeted at the Address-of-Record of bob (e.g. sip:bob@example.net). The "home" proxy for the domain of the original target, example.net, accesses the location service/registration database to determine the registered contact(s) for the user (e.g. sip:bob@192.168.0.10). This contact was retrieved in a REGISTER request from the user's UA. The incoming request is then re-targeted to the registered contact. When retarget-requests is enabled, or the original Request-URI is the Oracle Enterprise Session Border Controller itself, the URI from the LRT lookup is used as the new Request-URI for the outgoing request.

When a request is routed rather than re-targeted, the Request-URI is not changed, but one or more Route headers may be inserted into the outgoing request. Sometimes a request which already contains Route headers will be routed without adding additional Route headers.

When the Oracle Enterprise Session Border Controller routes requests and the original Request-URI was not the Oracle Enterprise Session Border Controller itself, the URI from the LRT /ENUM lookup is added as the top Route: header including the "lr" parameter. The Request-URI then remains unchanged.

Whether the Oracle Enterprise Session Border Controller re-targets or routes a request depends on the following:

- The target (Request-URI) of the received request
- The presence of Route headers
- Local Policy Attributes,
- Registration Cache matching.

If the original target is the Oracle Enterprise Session Border Controller itself (i.e. the Request-URI contains the IP Address in the SIP interface the request was received on), the request is always retargeted. When the original target is not the Oracle Enterprise Session Border Controller and Local Policy is applied, the request will be re-targeted when the policy attribute action parameter is replace-uri. The request will also be re-targeted when the policy attribute specifies an ENUM or LRT lookup.

Retargetting requests can be configured in either the ENUM or LRT config depending on the request URI retrieval method choosen.

Re-targeting LRT ENUM-based Requests Configuration

This section shows you how to configure the Oracle Enterprise Session Border Controller to retarget/re-route request message when performing an LRT or an ENUM lookup.

To configure the Oracle Enterprise Session Border Controller to retarget or route request messages when performing an LRT lookup:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-routing-config and press Enter.

```
ACMEPACKET(session-router)# local-routing-config
ACMEPACKET(local-routing-config)#
```

4. retarget-requests—Leave this parameter set to enabled for the Oracle Enterprise Session Border Controller to replace the Request-URI in the outgoing request. Change this parameter to disabled for the Oracle Enterprise Session Border Controller to route the request by looking to the Route header to determine where to send the message.
5. Save your work using the done command.

To configure the Oracle Enterprise Session Border Controller to retarget or route request messages when performing an ENUM lookup:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

7. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

8. Type local-routing-config and press Enter.

```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```

9. retarget-requests—Leave this parameter set to enabled for the Oracle Enterprise Session Border Controller to replace the Request-URI in the outgoing request. Change this parameter to disabled for the Oracle Enterprise Session Border Controller to route the request by looking to the Route header to determine where to send the message.
10. Save your work using the done command.

Recursive ENUM Queries

If the Oracle Enterprise Session Border Controller receives an A-record in response to an ENUM query, it will reperform that ENUM query to the server received in the A-record.

If the Oracle Enterprise Session Border Controller receives an NS record in response to an ENUM query, it will resend the original ENUM query to the DNS server defined in the realm of the FQDN in the NS record. It will use the response to perform a subsequent ENUM query.

This behavior is configured by setting the recursive query parameter in the enum config to enabled.

Recursive ENUM Queries Configuration

To configure the Oracle Enterprise Session Border Controller to query a DNS server for a hostname returned in an ENUM lookup result:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-routing-config and press Enter.

```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```

4. recursive-query—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to query a DNS server for a hostname returned in an ENUM result.
5. Save your work using the done command.

Multistage Local Policy Routing

Multistage local policy routing enables the Oracle Enterprise Session Border Controller to perform multiple stages of route lookups where the result from one stage is used as the lookup key for the next routing stage.

Routing Stages

A routing stage signifies a re-evaluation of local policy based on the results of a local policy lookup. In the simplest, single stage case, the Oracle Enterprise Session Border Controller performs a local policy lookup on a SIP message's Request URI. The result of that local policy lookup is a next hop FQDN, IP address, ENUM lookup, or LRT lookup; that result is where the Oracle Enterprise Session Border Controller forwards the message. In the multistage routing model, that resultant next hop is used as the lookup key for a second local policy lookup.

The results of each stage do not erase the results of the previous stage. Thus, previous results are also possible routes to use for recursion, but the next stage results are tried first.

Note: Setting a next hop to a SAG in a multistage scenario constitutes an error.

Multi-stage Routing Source Realm

By default, the Oracle Enterprise Session Border Controller uses the realm within which a message was received as the source realm through all stages of a multistage local policy routing lookup. You can change this by setting the multi-stage-src-realm-override parameter in the session router config to enabled. Enabling this setting causes the Oracle Enterprise Session Border Controller to use the next-hop realm from the current local policy stage as the source realm for the next stage of the lookup. This source realm selection process also repeats for each stage of a multistage routing scenario.

Network Applications

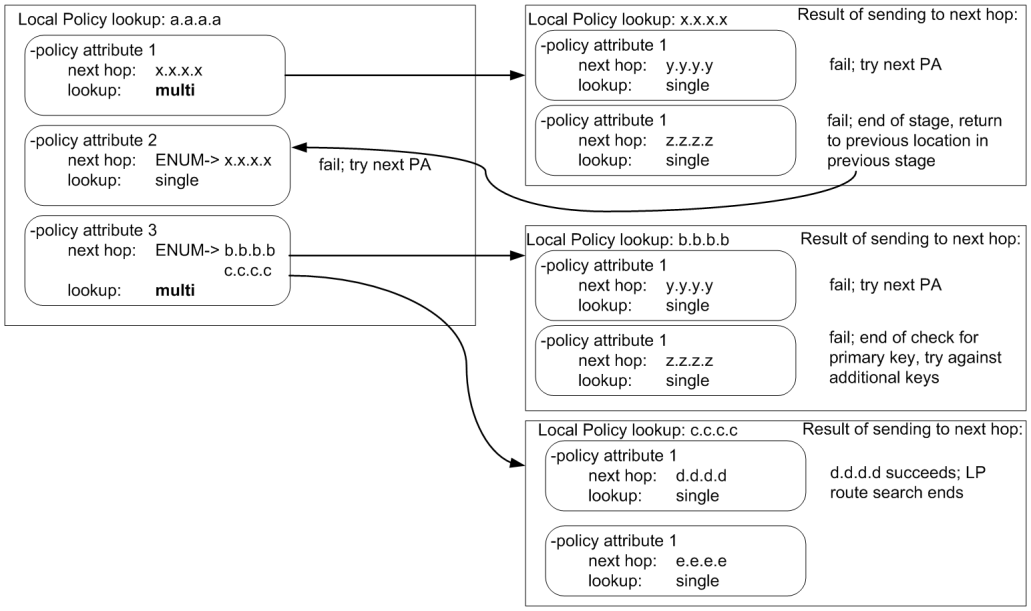
The following are typical applications of multistage routing:

- An operator might need to query an ENUM server for a destination number. Based on the NAPTR result of the ENUM query, the Oracle Enterprise Session Border Controller performs a local policy lookup to decide how to route the request, perhaps based on a LRT table lookup.
- An operator might need to query one ENUM server for a number portability lookup, then based on the routing number perform a second ENUM query to a different server to learn which carrier to use for the routing number. Then, then based on the identified carrier perform a LRT lookup for what next-hop(s) to use for that carrier.
- An operator might query an LRT table to confirm the allowed source number. Then, based on the result, query an ENUM server for destination routing.

Multistage Routing Conceptual Example

Multistage routing is enabled by setting a policy attribute's lookup parameter to multi. Instead of replacing the SIP message's request URI with the policy attribute's next hop address or response from an ENUM or LRT lookup, the system uses that next hop or ENUM or LRT lookup response to reconstruct the SIP message. The reconstructed SIP message is fed again through all configured local policy configuration elements (and policy attribute sub elements). Each time the Oracle Enterprise Session Border Controller re-evaluates a SIP message against local policies, it is considered an additional routing stage. When multiple records are returned from an ENUM or LRT lookup, the Oracle Enterprise Session Border Controller evaluates the first response against all applicable local policies. If unsuccessful, the Oracle Enterprise Session Border Controller evaluates all additional responses, in turn, against all applicable local policies.

For example:



Multistage Routing Example 2

The following three local policy configuration elements are configured in the Oracle Enterprise Session Border Controller:

Local Policy 1

from-addres	*
to-address	159
source-realm	private
policy-attribute	
next-hop	lrt:default-lrt
lookup	multi
policy-attribute	
next-hop	192.168.200.50
lookup	single

Local Policy 2

from-addres	*
to-address	192.168.1.49
source-realm	private
policy-attribute	
next-hop	lrt:carrier-lrt
lookup	multi
policy-attribute	
next-hop	lrt:emergency
lookup	single

Local Policy 3

from-addres	*
to-address	215680000002
source-realm	private
policy-attribute	
next-hop	192.168.200.98
lookup	single
policy-attribute	
next-hop	192.168.200.97
lookup	single
policy-attribute	
next-hop	192.168.200.44
lookup	multi

```
<route>
    <user type="E164">159</user>
        <next type="regex">!^.*$!sip:11568000000@192.168.200.47!</next>
        <next type="regex">!^.*$!sip:215680000002@192.168.200.99!</next>
        <next type="regex">!^.*$!sip:11578000000@192.168.200.44!</next>
</route>
```

```
INVITE sip:159@192.168.1.49:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.48:5060
From: sipp <sip:sipp@192.168.1.48:5060>;tag=1
To: sut <sip:159@192.168.1.49:5060>
Call-ID: 1-4576@192.168.1.48
CSeq: 1 INVITE
Contact: sip:sipp@192.168.1.48:5060
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:  135
```

The local route table in default-lrt appears as follows:

```
<route>
    <user type="E164">159</user>
                <next type="regex">!^.*$!sip:11568000000@192.168.200.47!</
next>
        <next type="regex">!^.*$!sip:215680000002@192.168.200.99!</next>
        <next type="regex">!^.*$!sip:11578000000@192.168.200.44!</next>
</route>
```

1. The Oracle Enterprise Session Border Controller receives an INVITE on realm, private (SDP is omitted below):

```
INVITE sip:159@192.168.1.49:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.48:5060
From: sipp <sip:sipp@192.168.1.48:5060>;tag=1
To: sut <sip:159@192.168.1.49:5060>
Call-ID: 1-4576@192.168.1.48
CSeq: 1 INVITE
Contact: sip:sipp@192.168.1.48:5060
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length:  135
```

2. The Oracle Enterprise Session Border Controller performs a local policy search based on the following parameters:

```
from-address:    sipp <sip:sipp@192.168.1.48:5060>;tag=1
to-address:            sip:159@192.168.1.49:5060
Source Realm:    private
```

3. The local policy search returns the four following routes to try:

```
lrt:default-lrt
192.168.200.50
lrt:emergency
lrt:carrier-lrt
```

The first next-hop route will be an LRT query. In addition, this policy attribute is configured with lookup=multi, meaning the results of the LRT query should be used for another local policy query, i.e., a second stage. More specifically, the request-uri that was received in response to the LRT query will be used as the to-uri in the next LP query.

The Oracle Enterprise Session Border Controller performs the LRT lookup in the default-lrt configuration element and is returned the following:

```
sip:11568000000@192.168.200.47
sip:215680000002@192.168.200.99
sip:11578000000@192.168.200.44
```

The Oracle Enterprise Session Border Controller attempts to use the results from the LRT query for the next stage Local Policy lookup(s). Beginning with the first route and continuing in sequential order, the Oracle Enterprise Session Border Controller will try to route the outgoing INVITE message by performing additional Local Policy lookups on the remaining LRT query results, until the INVITE is successfully forwarded.

The Oracle Enterprise Session Border Controller performs a local policy query on:

```
sip:11568000000@192.168.200.47
```

Which equates to a local policy lookup on:

```
        from-URI=sipp <sip:sipp@192.168.1.48:5060>;
        to-URI=sip:11568000000@192.168.200.47
Source Realm: private
```

The query fails because there is no Local Policy entry for 11568000000.

The Oracle Enterprise Session Border Controller performs a second query on request-uri

```
sip:215680000002@192.168.200.99
```

Which equates to a local policy lookup on:

```
        from-URI=sipp <sip:sipp@192.168.1.48:5060>;
        to-URI=sip:215680000002@192.168.200.99
Source Realm: private
```

The LP query is successful and returns the following next- hops:

```
192.168.200.98
    192.168.200.99
192.168.200.44
```

The three routes shown above represent the next stage of the multistage routing for this INVITE. The policy attributes' lookup parameter is set to single for these next-hops. Therefore, the Oracle Enterprise Session Border Controller will attempt to send the outgoing INVITE message to one or more of these next-hops; there are no more stages to check.

4. The Oracle Enterprise Session Border Controller sends an INVITE to 192.168.200.98:

```
INVITE sip:215680000002@192.168.200.98;lr SIP/2.0
Via: SIP/2.0/UDP 192.168.200.49:5060
From: sipp <sip:sipp@192.168.1.48:5060>
To: sut <sip:159@192.168.1.49:5060>
Call-ID: SDnhae701-76e8c8b6e168958e385365657faab5cb-v3000i1
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.49:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
```

5. If the INVITE is sent to 192.168.200.98 successfully, the local policy routing will conclude and the call will continue processing. Otherwise the Oracle Enterprise Session Border Controller will try the other next hops until a route succeeds or all next-hops have been exhausted

Customizing Lookup Keys

When the next hop parameter points to perform an ENUM or LRT lookup, it can be provisioned with a "key=" attribute in order to specify a parameter other than the username to perform the lookup on. The following table lists the header, key value, and corresponding syntax to configure the Oracle Enterprise Session Border Controller with.

Username from Header:	Key Value	Example
To-URI	$TO	key=$TO
From-URI	$FROM	key=$FROM
P-Asserted-Identity	$PAI	key=$PAI

For a subsequent stage in multistage local policy routing, the lookup key to use for the next stage can be explicitly specified by configuring the next key parameter. By default, multistage lookups use the modified Request-URI returned from the ENUM/LRT response as the to-address key for the next local policy lookup. When the next key parameter is configured, its value will be used for the to-address key in the subsequent local policy lookup regardless if an ENUM or LRT lookup is configured for that policy attribute. The key syntax is for this parameter is the same as with the Routing-based RN and CIC feature.

Multistage Routing Lookup Termination

It is important for the Oracle Enterprise Session Border Controller to have a mechanism to stop performing additional stages of route lookups and limit the number of attempts and results to be tried. Routing termination can be performed at in the non-multistage way or at the global session router level.

Global Local Policy Termination

The Oracle Enterprise Session Border Controller can be configured to limit local policy lookups based several aspects of the route lookup process:

- Limiting the number of stages per message lookup—The Oracle Enterprise Session Border Controller can limit to the number of additional local policy lookup stages it will perform received message to a maximum of 5. This is configured with the additional lp lookups parameter. Leaving this parameter at its default value of 0 essentially disables multistaged local policy lookups.
- Limiting the number of routes per Local Policy lookup—The Oracle Enterprise Session Border Controller can limit the number of route results to use as returned for each Local-Policy lookup. This is configured with the max

lp lookups routes per lookup parameter. Leaving this parameter at its default value of 0 places no limit on the number of returned routes the Oracle Enterprise Session Border Controller can try.

- Limiting the total number of routes for all local policy lookups per message request—The Oracle Enterprise Session Border Controller can limit the number of route returned in total across all lookups for a given request, including additional stages. This is configured with the total lp routes parameter. Leaving this parameter at its default value of 0 places no limit on the number of returned routes the Oracle Enterprise Session Border Controllercan try. This parameter overrides any configured options.

Additionally, the Oracle Enterprise Session Border Controller monitors for local policy lookup loops which could cause a significant deterioration in performance. If a loop is found, the Oracle Enterprise Session Border Controller stops trying the looping route list and proceeds to try any remaining routes..

Multistage Local Policy Routing Configuration

To set up your local policy attributes for routing using the TO header:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type local-policy and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit a local policy.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type policy-attributes and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

5. next-hop—This is the next signaling host and/or object to query. This parameter can be configured as an IP address, ENUM server, or LRT. You can also add a lookup key to an ENUM server or LRT lookup with the following syntax:

```
next-hop    enum:ENUM-object;key=$TO
```

6. terminate-recursion—Set this parameter to enabled to terminate local policy route recursion when the current stage completes.
7. lookup—Leave this parameter at the default single for single stage local policy routing or set it to multi to enable multistage local policy routing.
8. next-key—Set this parameter to $TO, $FROM, or $PAI if you wish to override the recently-returned lookup key value for the next stage.
9. Save and activate your configuration.

Maintenance and Troubleshooting

The show sipd policy command includes four additional counters that refer to single and multistage local policy lookups. All counters are reported for the recent period, and lifetime total and lifetime period maximum. These counters are:

- Local Policy Inits—Number of times the Oracle Enterprise Session Border Controller makes an initial local policy lookup.
- Local Policy Results Max—Number of times the Oracle Enterprise Session Border Controller truncated the number of routes returned for a local policy lookup because the maximum number of routes per local policy lookup (max lp lookups routes per lookup) threshold was reached.

- Local Policy Exceeded—Number of times the Oracle Enterprise Session Border Controller truncated the number of routes returned for a local policy lookup because the maximum number of routes per message request (total lp routes) threshold was reached.
- Local Policy Loops—Number of times the Oracle Enterprise Session Border Controller detected a loop while performing a multistage local policy lookup.

Traps

An SNMP trap is generated to notify that the limit on the additional lp lookups threshold has been reached during the recent window period. This trap occurs a maximum of once during a window period.

```
apSysMgmtLPLookupExceededTrap NOTIFICATION-TYPE
    STATUS           current
       DESCRIPTION
          " The trap will be generated the first time the additional Local
Policy Lookups limit is reached is in the recent window period. The trap will
only occur once during a window period."
       ::= { apSystemManagementMonitors 65}
```

Routing-based RN and CIC

When the Oracle Enterprise Session Border Controller performs local policy routing, it selects local policy entries based on from addresses, to addresses, and source realms. All three are configurable in the local policy configuration. The to addresses can either be the username in a Request-URI (if it is an E.164/phone number format), or the request-URI's hostname or IP address. The Oracle Enterprise Session Border Controller sorts matching local policies based on policy attribute entries. A policy attribute defines a next hop, which can be a session agent or a session agent group. Alternatively, the next hop might define an ENUM server group or local route table to use to find the next hop.

If the routing-based RN and CIC feature is not enabled, the Oracle Enterprise Session Border Controller performs the subsequent ENUM query or local route table lookup using the Request-URI's username, if it is a telephone number (TN). The TN is the normalized user part of the Request-URI, ignoring any user parameters or non-digit characters.

If the routing-based RN and CIC feature is enabled, the Oracle Enterprise Session Border Controller instead performs the ENUM or local route table lookup based on a user parameter, which is useful for lookups based on routing number (RN) or carrier identification code (CIC):

- An RN is a number that identifies terminating switch nodes in Number Portability scenarios when the original TN has been moved to the switch defined by the RN.
- A CIC is the globally unique number of the terminating carrier to which a ported number has been moved.

In applications where the Oracle Enterprise Session Border Controller is given the RN or the CIC in the Request-URI, this feature is useful because the Oracle Enterprise Session Border Controller can perform an additional ENUM or local route table lookup to find the next hop to the RN or the CIC. Typically, ENUM servers have imported Number Portability data with which to respond to the Oracle Enterprise Session Border Controller query, and (for example) the Oracle Enterprise Session Border Controller can use local route tables for storing CIC values for direct carrier hand-off.

Even with this feature enabled, the Oracle Enterprise Session Border Controller still performs local policy match selection based on the TN. This feature only uses the RN or CIC user-parameter for the ENUM or local route table lookup after the local policy and policy attributes have been selected.

Routing-based RN Configuration

This section shows you how to specify that a set of local policy attributes should use an RN for lookup. You can also set this value to CIC, or to any value you require.

You can set the lookup key to an RN in the local policy attributes' next-hop parameter.

To set the lookup key to RN:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-policy and press Enter.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type policy-attributes and press Enter.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

5. next-hop—In the next-hop parameter—after the kind of ENUM service used—type a colon (;). Then, without spaces, type in key=rn and press Enter.

```
ACMEPACKET(local-policy-attributes)# next-hop lrt:lookup;key=rn
```

6. Save and activate your configuration.

Codec Policies for SIP

The Oracle Enterprise Session Border Controller has the ability to add, strip, and reorder codecs for SIP sessions. This builds on the Oracle Enterprise Session Border Controller's pre-existing abilities to route by codec and reorder one codec in an SDP offer by allowing you to configure the order of multiple codecs and to remove specific codecs within the media descriptions in SDP offers.

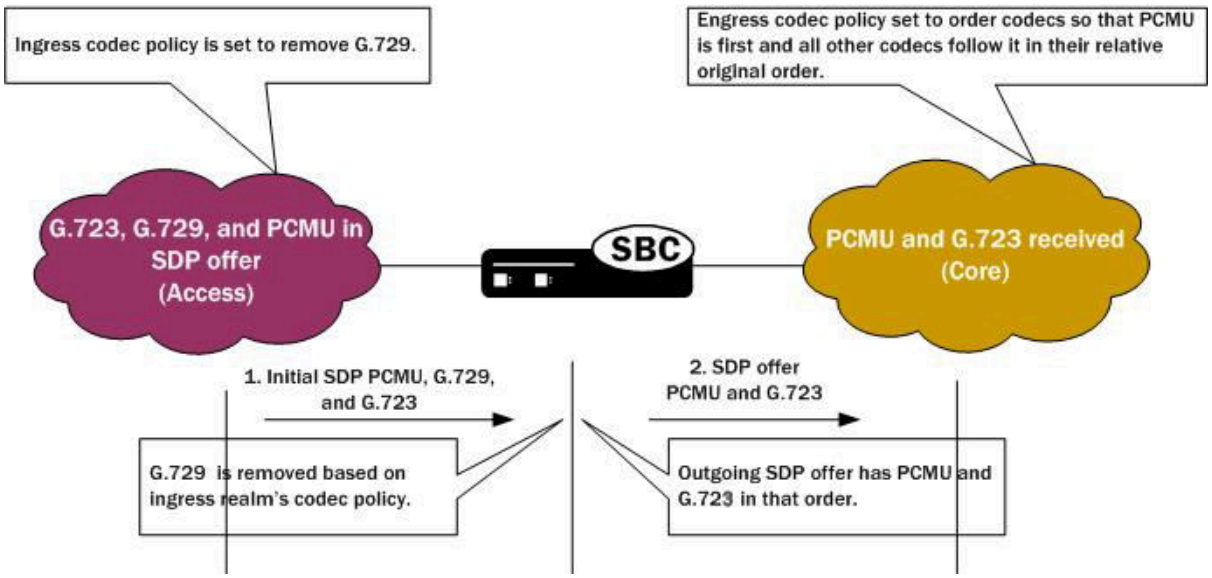
You can enable the Oracle Enterprise Session Border Controller to perform these operations on SDP offers by configuring codec policies. Codec policies are sets of rules that specify the manipulations to be performed on SDP offers. They are applied on an ingress and egress basis using the realm and session agent configurations.

Oracle Enterprise Session Border Controller supports three types of codec policies:

- Ingress policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for incoming traffic
- Egress policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for traffic leaving the Oracle Enterprise Session Border Controller
- Conditional policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for traffic leaving the Oracle Enterprise Session Border Controller. A conditional policy differs from an egress policy in providing the capability to perform standard codec manipulations (add and strip) dynamically, based on the codec list and associated parameters contained in the original SDP offer.

The Oracle Enterprise Session Border Controller applies codec policies during the offer phase of media format negotiation. If codec manipulation is enabled, then the Oracle Enterprise Session Border Controller performs the modification according to the specific policy and forwards on the traffic.

For example, when the Oracle Enterprise Session Border Controller receives a SIP INVITE with SDP, it refers to the realm through which the INVITE arrived and performs any manipulations specified by an ingress codec policy that may have been assigned to the ingress realm. With the media description possibly changed according to the ingress codec policy, the Oracle Enterprise Session Border Controller passes the SDP offer to the outgoing realm so that the an egress codec policy can be applied. Note that the SDP to be evaluated by the egress codec policy may match the original SDP, or it may have been changed during transit through the ingress realm. After applying the egress coded policy, the Oracle Enterprise Session Border Controller forwards the INVITE.



Since the offer-answer exchange can occur at different stages of SIP messaging, the assigned ingress and egress roles follow the media direction rather than the signaling direction. It might be, for example, that the offer is in an OK that the Oracle Enterprise Session Border Controller modifies.

You can apply codec policies to realms and to session agents; codec policies configured in session agents take precedence over those applied to realms. However, it is not required that there be both an ingress and an egress policy either for realms or for session agents. If either one is unspecified, then no modifications take place on that side. If neither ingress nor egress policies specified, SDP offers are forwarded as received.

Relationship to Media Profiles

For each codec that you specify in a codec policy, there must be a corresponding media profile configuration on the Oracle Enterprise Session Border Controller. You configure media profiles in the ACLI via the session-router path. In them, you can specify codec type, transport protocol, required bandwidth, and a number of constraints.

Manipulation Modes

You can configure a codec policy to perform several different kinds of manipulations:

- Allow—List of codecs that are allowed for a certain codec policy; if a codec does not appear on this list, then the Oracle Enterprise Session Border Controller removes it. You can wildcard this list with an asterisk (*) so that all codecs are allowed. Further, you can create exceptions to a wildcarded allow list.
  - You make an exception to the wildcarded list of codecs by entering the codec(s) that are not allowed with a no attribute. This tells the Oracle Enterprise Session Border Controller to allow all codecs except the one(s) you specify.

    ```
    ACMEPACKET(codec-policy)# allow-codecs (* PCMA:no)
    ```
  - You can also create exceptions to allow lists such that audio or video codecs are removed. However, when the allow list specifies the removal of all audio codecs and an INVITE arrives at the Oracle Enterprise Session Border Controller with only audio codecs, the Oracle Enterprise Session Border Controller behaves in accordance with RFC 3264. This means that the resulting SDP will contain one attribute line, with the media port for the media line set to 0. The terminating side will need to supply new SDP in its reply because the result of the manipulation is the same as an INVITE with no body.

    ```
    ACMEPACKET(codec-policy)# allow-codecs (* audio:no)
    ```
- Order—List of the codecs where you specify their preferred order in the outgoing media offer. The Oracle Enterprise Session Border Controller arranges matching codecs according to the rule you set, and any remaining ones are added to the list in the same relative order they took in the incoming media offer. If your list specifies a codec that is not present, then the ordering proceeds as specified but skips the missing codec.

You can use an asterisk (*) as a wildcard in this list, too. The placement of the asterisk is key, as you can see in the following examples:

- For an order rule set this way

  ```
  ACMEPACKET(codec-policy)# order (A B C *)
  ```

  codecs A, B, and C will be placed at the front of the codec list in the order specified; all other codecs in the offer will follow A, B, and C in the same relative order they had in the original SDP offer.
- For an order rule set this way:

  ```
  ACMEPACKET(codec-policy)# order (* A B C)
  ```

  codecs A, B, and C will be placed at the end of the codec list in the order specified; all other codecs in the offer will come before A, B, and C in the same relative order they had in the original SDP offer.
- For an order rule set this way

  ```
  ACMEPACKET(codec-policy)# order (A * B C)
  ```

  codec A will be placed at the beginning of the codec list, to be followed by all other codecs in the offer in the same relative order they had in the original SDP offer, and then B and C will end the list.

- Force—An attribute you can use in the allow list with one codec to specify that all other codecs should be stripped from the outgoing offer. You can specify multiple forced codecs in your rules.
  - If you set multiple codecs in the allow list and one of them is forced, then the outgoing offer will contain the forced codec.
  - If you set multiple codecs in the allow list and the one that is forced is not present in the offer, then the Oracle Enterprise Session Border Controller will select a non-forced codec for the outgoing offer.

  ```
  ACMEPACKET(codec-policy)# allow (PCMU G729:force)
  ```

  You cannot use the force attribute with a wildcarded allow list.
- No—An attribute that allows you to strip specified codecs or codec types from a wildcarded allow list.

  ```
  ACMEPACKET(codec-policy)# allow (* PCMA:no)
  ```

In-Realm Codec Manipulation

In addition to being able to apply codec policies in realms, the realm configuration supports a setting for determining whether codec manipulation should be applied to sessions between endpoints in the same realm.

In-realm codec manipulation can be used for simple call flows that traverse two realms. If the originating and terminating realms are the same, the Oracle Enterprise Session Border Controller checks to see if you have enabled this capability. If you have enabled it, then the Oracle Enterprise Session Border Controller performs the specified manipulations. If this capability is not enabled, or if the realm's media management in realm (mm-in-realm) setting is disabled, then the Oracle Enterprise Session Border Controller does not perform codec manipulations.

For more complex calls scenarios that involve call agent or reinitiation of a call back to the same realm, the Oracle Enterprise Session Border Controller does not perform in-realm codec manipulation.

Codec Policy Configuration

This section gives instructions and examples for how to configure codec policies and then apply them to realms and session agents. It also shows you how to configure settings for in-realm codec manipulation.

Creating a Codec Policy

To create a codec policy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type media-manager and press Enter to access the signaling-related configurations.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type codec-policy and then press Enter.

```
ACMEPACKET(media-manager)# codec-policy
ACMEPACKET(codec-policy)#
```

4. name—Enter the unique name for the codec policy. This is the value you will use to refer to this codec policy when you apply it to realms or session agents. This parameter is required and is empty by default.
5. allow-codecs—Enter the list of media format types (codecs) to allow for this codec policy. In your entries, you can use the asterisk (*) as a wildcard, the force attribute, or the no attribute so that the allow list you enter directly reflects your configuration needs. Enclose entries of multiple values in parentheses (( )).

   The codecs that you enter here must have corresponding media profile configurations.
6. add-codecs-on-egress—Enter the codecs that the Oracle Enterprise Session Border Controller adds to an egress SDP offer if that codec is not already there. This parameter applies only to egress offers.

   The codecs that you enter here must have corresponding media profile configurations.

   add-codecs-on-egress can be used to construct ingress, egress, or conditional codec policies.
7. order-codecs—Enter the order in which you want codecs to appear in the outgoing SDP offer. Remember that you can use the asterisk (*) as a wildcard in different positions of the order to directly reflect your configuration needs. Enclose entries of multiple values in parentheses (( )).

   The codecs that you enter here must have corresponding media profile configurations.
8. Save and activate your configuration.

   Your codec policy configuration will resemble the following example:

```
codec-policy
        name            private
        allow-codecs    g723:no pcmu video:no
        order-codecs    pcmu
```

Applying a Codec Policy to a Realm

Note that codec policies defined for session agents always take precedence over those defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.
4. codec-policy—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

Applying a Codec Policy to a Session Agent

Note that codec policies that are defined for session agents always take precedence over those that are defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter.

```
ACMEPACKET(session-router)# session-agent
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the realm that you want to edit.

4. codec-policy—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

In-Realm Codec Manipulations

To enable in-realm codec manipulations:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
```

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. codec-manip-in-realm—Enter the name of the codec policy that you want to apply to this realm. The default value is disabled. The valid values are:
   - enabled | disabled
5. Save and activate your configuration.

QoS Based Routing

In addition to configuring your system for routing based on certain session constraints, you can also set up routing based on QoS. QoS based routing uses the R-Factor on a per-realm basis to either cut back on the traffic allowed by a specific realm, or to shut that traffic off altogether.

To use this feature, you set up QoS constraints configurations and apply one per realm. The QoS constraints configuration allows you to set up two thresholds:

- Major—The major threshold sets the R-Factor limit beyond which the Oracle Enterprise Session Border Controller rejects a certain percentage (that you configure) of calls. That is to say, it rejects inbound calls at the rate you set with a 503 Service Unavailable status code, and rejects outbound calls if there are no alternative routes.
- Critical—The critical threshold, when exceeded, causes the Oracle Enterprise Session Border Controller to behave the same way it does when any of the session constraints (set in the session-constraints configuration) are exceeded. All inbound calls to the realm are rejected with a 503 Service Unavailable status code, and (if there is no alternate route) outbound calls are rejected, too. Until the R-Factor falls within acceptable means and the session constraint's time-to-resume value has elapsed, the realm remains in this state.

Management

This feature is supported by MIBs and traps. Historical data recording (HDR) also supports this feature by providing the following metrics in the session realm statistics collection group:

- Average QoS RFactor (0-93)

- Maximum QoS RFactor (0-93)
- Current QoS Major Exceeded
- Total QoS Major Exceeded
- Current QoS Critical Exceeded
- Total QoS Critical Exceeded

QoS Contraints Configuration

This section shows you how to configure a QoS constraints configuration and then how to apply it to a realm.

Configuring QoS Constraints

Your first step to enabling QoS based routing is to set up a QoS constraints configuration. This configuration is where you enter major and critical thresholds, as well as the load reduction for the realm should the R-Factor exceed the major threshold.

To set up a QoS constraints configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type qos-constraints and press Enter.

```
ACMEPACKET(session-router)# qos-constraints
ACMEPACKET(qos-constraints)#
```

4. name—Enter the name of this QoS constraints configuration. This parameter uniquely identifies the configuration, and you use this value when applying the configuration to a realm. This parameter has no default and is required.
5. state—Set the state of this QoS constraints configuration. The default is enabled, but you can set this parameter to disabled if you want to stop applying these constraints.
6. major-rfactor—Enter a numeric value between 0 (default) and 9321 to set the threshold that determines when the Oracle Enterprise Session Border Controller applies the call reduction rate. If you leave this parameter set to 0, then the Oracle Enterprise Session Border Controller will not apply a major threshold for any realm where you apply this QoS constraints configuration.

   Note that this value must be greater than that you set for the critical-rfactor, except when the major-rfactor is 0.
7. critical-rfactor—Enter a numeric value between 0 (default) and 9321 to set the threshold that determines when the Oracle Enterprise Session Border Controller rejects all inbound calls for the realm, and rejects outbound calls when there is no alternate route. If you leave this parameter set to 0, then the Oracle Enterprise Session Border Controller will not apply a critical threshold for any realm where you apply this QoS constraints configuration.

   Note that this value must be less than that you set for the major-rfactor, except when the major-rfactor is 0.
8. call-load-reduction—Enter a number from 0 (default) to 100 representing the percentage by which the Oracle Enterprise Session Border Controller will reduce calls to the realm if the major-rfactor is exceeded. If you leave this parameter set to 0, then the Oracle Enterprise Session Border Controller will not reduce call load for the realm —even when the major-rfactor is configured.

   This is the percentage of inbound and outbound calls the Oracle Enterprise Session Border Controller will reject. For example, if you set this parameter to 50 and the major threshold is exceeded, then the Oracle Enterprise Session Border Controller rejects every other call to the realm.
9. Save and activate your configuration.

Applying QoS Constraint to a Realm

You apply QoS constraints to realms using the qos-constraint parameter.

To apply a QoS constraint to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. If you adding this feature to a pre-existing realm, then you need to select and edit that realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. qos-constraints—Enter the name value from the QoS constraints configuration you want to apply to this realm.

   Save and activate your configuration.

14

Using the Local Route Table (LRT) for Routing

The LRT allows the Oracle Enterprise Session Border Controller to determine next hops and map E.164 to SIP URIs locally for routing flexibility.

The LRT uses a local route cache that is populated by a local XML file on the Oracle Enterprise Session Border Controller. Each local cache is populated from one defined XML file. For routing, the local route cache operates in a way similar to the ENUM model where a local policy next hop specifies the local route table that the Oracle Enterprise Session Border Controller references. For example, you can configure one next hop to use one table, and another next hop to use a different table.

Similar to the ENUM model, the Oracle Enterprise Session Border Controller typically performs a local route table lookup using the telephone number (TN) of the SIP Request-URI. This is the user portion of the URI, and the Oracle Enterprise Session Border Controller ignores user parameters or non-digit characters. The local route table XML file defines the matching number and the resulting regular expression replacement value such as ENUM NAPTR entries do. The Oracle Enterprise Session Border Controller uses the resulting regular expression to replace the Request-URI, and it uses the hostname or IP address portion to determine the next hop. If the hostname or IP address matches a configured session agent, the request is sent to that session agent. If the Oracle Enterprise Session Border Controller does not find a matching session agent for the hostname/IP address, the Oracle Enterprise Session Border Controller either performs a DNS query on the hostname to determine its IP address or sends the request directly to the IP address.

When the next hop is defined as a user-parameter lookup key, such as a routing number (RN) or carrier identification code (CIC), the defined key is used for the local route table lookup.

The Oracle Enterprise Session Border Controller can attempt up to 10 next hops per LRT entry in the order in which they appear in the XML file. If the next hop is unsuccessful, the Oracle Enterprise Session Border Controller tires the next hop on list. An unsuccessful hop may occur when an out-of-service session agent or the next hop responds with a failure response.

Note: Entering XML comments on the same line as LRT XML data is not supported.

The Oracle Enterprise Session Border Controller can perform local route table lookups for SIP requests and communicate the results to the SIP task. The new task processes the new local routing configuration objects.

When a SIP call is routed, the Oracle Enterprise Session Border Controller uses local policy attributes to determine if a local route table lookup is required. If a lookup is needed, the Oracle Enterprise Session Border Controller selects the local routing configuration to use. Successful local route table lookups result in URIs that can be used to continue routing and redirecting calls.

Local Route Table (LRT) Performance

Capabilities

- Loads approximately 500 LRT tables during boot time
- Loads 100,000 entries per LRT file
- Loads 2,000,000 LRT entries total per system

Constraints

- You cannot configure the Oracle Enterprise Session Border Controller with 500 LRT files each with 100,000 entries.
- Actual performance that affects the interaction among the three performance attributes varies with system memory and configuration.

Local Routing Configuration

This section shows you how to:

- Set up local route configuration
- Specify that a set of local policy attributes needs to use local routing

Configure Local Routing

The local routing configuration is an element in the ACLI session-router path, where you configure a name for the local route table, the filename of the database corresponding to this table, and the prefix length (significant digits/bits) to be used for lookup.

To configure local routing:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-routing-config,and press Enter.

```
ACMEPACKET(session-router)# local-routing-config
ACMEPACKET(local-routing-config)#
```

4. name—Enter the name (a unique identifier) for the local route table; this name is used for reference in the local policy attributes when to specify that local routing should be used. There is no default for this parameter, and it is required.
5. file-name—Enter the name for the file from which the database corresponding to this local route table will be created. You should use the .gz format, and the file should be placed in the /code/lrt/ directory. There is no default for this parameter and it is required.
6. prefix-length—Enter the number of significant digits/bits to used for lookup and cache storage. The default value is 0. The valid range is:
   - Minimum—0
   - Maximum—999999999
7. Save and activate your configuration.

   The following example displays a typical local routing configuration.

```
local-routing-config
        name                            lookup
        file-name                       abc.xml.gz
        prefix-length                   3
```

Applying the Local Routing Configuration

Apply the local routing configuration by calling it to use in the local policy attributes. You do this by setting a flag in the next-hop parameter along with the name of the local routing configuration that you want to use.

To apply the local routing configuration:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type local-policy, and press Enter.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type policy-attributes, and press Enter.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

5. next-hop—In the next-hop parameter, type in lrt: followed directly by the name of the local routing configuration to be used. The lrt: tag tells the Oracle Enterprise Session Border Controller that a local route table will be used.

```
ACMEPACKET(local-policy-attributes)# next-hop lrt:lookup
```

6. Save and activate the configuration.

Local Route Table Support for H.323 and IWF

Local Route Table (LRT) support for H.323 and IWF is compatible with that currently offered for SIP. LRT and ENUM provide the Oracle Enterprise Session Border Controller with the ability to perform routing based on ENUM queries to a DNS server or local to an onboard database.

For the LRT feature, this means that entries in the local routing table now include those prefixed with the h323: URI scheme, indicating that H.323 is the next hop protocol.

IWF Considerations

When the system performs a local policy lookup for an incoming SIP or H.323 call and determines an ENUM/LRT server is the next hop, it queries that ENUM/LRT server. The response will include the URI scheme, indicating the next hop protocol and the hostname/IP address representing the next hop. For cases where the incoming call signaling protocol and the URI scheme of the ENUM/LRT response are the same, the call requires no interworking. The Oracle Enterprise Session Border Controller can simply route the egress call leg to the specified next hop.

Interworking is required when the incoming signaling protocol and the URI scheme of the ENUM/LRT response do not match. When the responses do not match, the Oracle Enterprise Session Border Controller interworks between SIP and H.323 to route the call to the appropriate next hop.

The Oracle Enterprise Session Border Controller also compares the URI scheme returned in the ENUM/LRT response to the application protocol specified in the policy attributes. If the URI scheme is SIP, but the policy attributes indicate H.323, the route is deemed invalid. The same is true for an H.323 URI scheme and SIP route.

ENUM LRT Responses

No special configuration is required for LRT to work for H.323 and IWF calls. You can configure the system to match ENUM/LRT responses against session agent groups, and then use those SAGs for routing.

To enable matching ENUM/LRT responses for H.323 SAG routing:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type h323-config, and press Enter.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

4. enum-sag-match—Set this parameter to enabled if you want the Oracle Enterprise Session Border Controller to perform matching against the hostnames in ENUM/LRT lookup responses and session agent groups. If there is a match, the Oracle Enterprise Session Border Controller uses the matching SAG for routing. If no match is found, normal ENUM/LRT routing proceeds.

15

Number Translation

About Number Translation

Oracle Enterprise Session Border Controller number translation is used to change a layer-5 endpoint name according to prescribed rules. Number translations can be performed on both the inbound and the outbound call legs independently, before and after routing occurs. Number translation is used for SIP, H.323, and SIP/H.323 interworking configurations.

Number translation takes place twice for both H.323 and SIP calls. The first number translation is applied to the incoming leg of the call, before the outgoing route is selected. The second number translation is applied to the outgoing leg of the call after the outgoing route is selected.

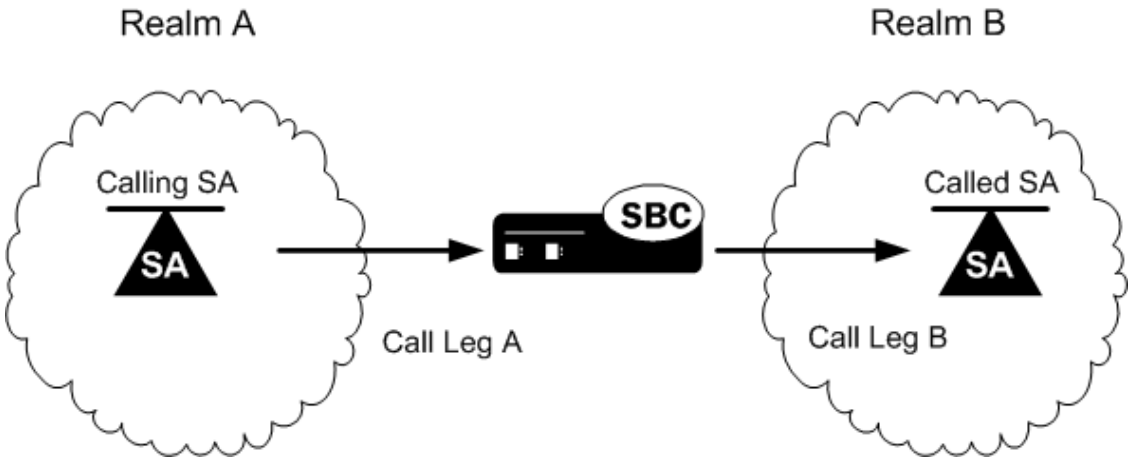
Number translation can be used to strip address prefixes added by external gateways. It can also be used to add a string tag to an address in order to implement a local policy routing scheme, and then remove the tag upon egress from the Oracle Enterprise Session Border Controller. The most common use of number translation is to add or remove a “1” or a + from a phone number sent from or addressed to a device.

Number Translation Implementation

Oracle Enterprise Session Border Controller number translations are implemented in three steps. First, the individual number translation rules are defined in the translation rules subelement. Next, the established rules are grouped in a specified order to apply to calling and called numbers. This second step occurs in the session translation element. Finally, session translations are attached to either session agents or realms in the session agent element or realm configuration element.

Number translations attached to session agents take precedence over number translations attached to realms. If no number translation is applied to a session agent, then the Oracle Enterprise Session Border Controller will use the number translation applied to a realm. If a number translation is applied to both a realm and session agent, the translation attached to the session agent will apply. If session agents and realms have no associated translations, then all numbers will remain in their original forms as they pass through the Oracle Enterprise Session Border Controller.

Within each realm or session agent, the number translation is applied to either the incoming or outgoing call leg. This distinction between incoming and outgoing calls is made from the point of view of the Oracle Enterprise Session Border Controller. The following diagram illustrates the number translation concept.



The following table shows you which parameters to apply a session translation ID in order to affect the corresponding leg of the call as shown in the illustration.

Leg	Calling SA	Called SA	Realm A	Realm B
A	IN Translation ID		IN Translation ID	
B		OUT Translation ID		OUT Translation ID

Number Translation in SIP URIs

Number translations only change the user portion of the URI. A typical SIP URI looks like sip:user@hostname. The user portion can take the form of either a phone number or any other string used for identification purposes.

Within the SIP header exists a Request URI, a To URI, and a From URI. The session translation element's rules calling parameter modifies the From URI, while the rules called parameter modifies the Request URI and the To URI.

Session Translation in H.323 Messages

Because H.323 messages explicitly define the calling and called parties, the correspondence is exactly the same between the endpoints and configuration parameters. The H.323 calling party corresponds to the session translation element's rules calling parameter. The H.323 called party corresponds to the session translation element's rules called parameter.

Number Translation Configuration Overview

Configuring the number translation feature requires the following steps:

1. Configure individual translation rules in the translation rules element.
2. Group these rules for use in the session translation element.
3. Apply these groups of rules on a per session agent or per realm basis using the appropriate fields in the session agent or realm configuration elements.

Translation Rules

The translation rules subelement is where the actual translation rules are created. The fields within this element specify the type of translation to be performed, the addition or deletion to be made, and where in the address that change takes place. Translations are not applied to any realm or session agent in this element.

When creating translation rules, first determine the type of translation to perform. The following table lists and describes the three types of number translations.

Field Value	Description
add	This translation type adds a character or string of characters to the address.
delete	This translation type deletes a character or string of characters from the address.
replace	This translation type replaces a character or string of characters within the address. Replace works by first applying the delete parameter then by applying the add parameter.

After you set the translation type, you define the string to add or delete. The wildcard term for a string to delete is the at-sign, @. Finally, you specify the character position in the address to make the addition or deletion.

The character position where an add or delete occurs is called an index. The index starts at 0 (immediately before the leftmost character) and increases by 1 for every position to the right you move. In order to specify the final position in an address, use the dollar-sign, $.

To create a translation rule that deletes a string:

Translation Rules for Deleting Strings

To create a translation rule that deletes a string:

1. Enter a descriptive name for this translation in the ID field.
2. If you are deleting a specific string, enter it in the delete string field.
3. If you are deleting a portion of the address string, enter the index number in the delete index field. For this type of deletion, remember to enter the number of characters you are deleting in the form of at-signs @ in the delete string field.

   The first matched string will be deleted, any remaining strings that match will remain. For example, if the address is 187521865 and the string to delete is “18,” only the first instance of 18 will be deleted. The second instance will remain after translation.

Translation Rules for Adding Strings

To create a translation rule that adds a string:

1. Enter a descriptive name for this translation in the ID field.
2. Enter the string you want to add in the add string field.
3. Enter the index of the string insertion in the add-index parameter. If you want to add a string at the end of an number, enter a dollar-sign $ in the add index field.

Translation Rules for Replacing Strings

To create a translation rule that replaces a string:

Note: A string replacement involves deleting a string followed by adding a string in the removed string’s place. The index is not used when replacing a string.

1. Enter a descriptive name for this translation in the ID field.
2. Enter the string you want to delete in the delete string field.
3. Enter the string you want to add in the add string field.

Session Translation

A session translation defines how translation rules are applied to calling and called numbers. Multiple translation rules can be referenced and applied using this element, which groups rules together and allows them to be referenced by one identifier.

There are two parameters in the session translation element. The rules calling parameter lists the translation rules to be applied to the calling number. The rules called parameter lists of translation rules to be applied to the called number.

The Oracle Enterprise Session Border Controller applies the translation rules in the order in which they are entered. They are applied cumulatively. For example, if this field is configured with a value of rule1 rule2 rule3, rule1 will be applied to the original number first, rule2 second, and rule3 last.

To configure the session translation element:

1. Enter a descriptive name for this session translation in the ID field.
2. Enter the IDs of existing translation rules in the rules calling parameter. Multiple rules can be entered in this field. The order you enter them in is the order in which they are applied.
3. Enter the IDs of existing translation rules in the rules called parameter. Multiple rules can be entered in this field. The order you enter them in is the order in which they are applied.

Applying Session Translations

Session translations can be applied to both session agents and realms. Both session agents and realms contain the two parameters that denote incoming and outgoing call legs—in translation ID and out translation ID. These two fields are populated with session translation element IDs.

If none of these fields are populated, no number translation will take place and the original address will remain unchanged as it traverses the Oracle Enterprise Session Border Controller. Further, any session translation applied to a session agent takes precedence over one applied to a realm.

Session Agent

To configure number translation for a session agent:

In the session agent element, set the in translation ID and/or the out translation ID to the appropriate ID you configured in the session translation element. There can be only one entry in each of these fields.

Realm

To configure number translation for a realm:

In the realm configuration element, set the in translation ID and/or the out translation ID to the appropriate ID you configured in the session translation element. There can be only one entry in each of these fields.

Number Translation Configuration

To create a translation rule:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type translation-rules and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# translation-rules
ACMEPACKET(translation-rules)#
```

From this point, you can configure translation rules parameters. To view all translation rules parameters, enter a ? at the system prompt. The following is an example of what a translation rule configuration might look like. Parameters not described in this section are omitted below.

```
translation-rules
        id                              addplus1
        type                            add
        add-string                      +1
        add-index                       0
        delete-string
        delete-index                    0
```

Translation Rules

Set the following parameters to configure a translation rule:

1. ID—Set a descriptive ID name for this translation rule.
2. type—Set the type of translation rule you want to configure. The default value is none. The valid values are:
   - add—Adds a character or string of characters to the address
   - delete—Deletes a character or string of characters from the address
   - replace—Replaces a character or string of characters within the address
   - none—Translation rule is disabled
3. add-string—Enter the string to be added during address translation to the original address. The value in this field should always be a real value; i.e., this field should not be populated with at-signs (@) or dollar-signs ($). The default value is a blank string.
4. add-index—Enter the position, 0 being the left most position, where you want to add the string defined in the add-string parameter. The default value is zero (0). The valid range is:
   - Minimum—0
   - Maximum—999999999
5. delete-string—Enter the string to be deleted from the original address during address translation. Unspecified characters are denoted by the at-sign symbol (@).

   The @ character only works if the type parameter is set to delete. This parameter supports wildcard characters or digits only. For example, valid entries are: delete-string=@@@@@, or delete-string=123456. An invalid entry is delete-string=123@@@. When the type is set to replace, this value is used in conjunction with the add-string value. The value specified in the delete-string field is deleted and the value specified in the add-string field is inserted. If no value is specified in the delete-string parameter and the type field is set to replace, then nothing will be inserted into the address. The default value is a blank string.
6. delete-index—Enter the position, 0 being the left most spot, where you want to delete the string defined in the delete-string parameter. This parameter is only used if the delete-string parameter is set to one or more at-signs. The default value is zero (0). The valid range is:
   - Minimum—0
   - Maximum—999999999

Session Translation

To configure session translations:

1. Exit out of the translation rules element and enter the session translation element.

   ```
   ACMEPACKET(translation-rules)# exit
   ACMEPACKET(session-router)# session-translation
   ACMEPACKET(session-translation)#
   ```

   From this point, you can configure the session translation element. To view all session translation parameters, enter a ? at the system prompt. The following is an example of what a session translation configuration might look like:

   ```
   session-translation
           id                              lrules-out
           rules-calling                   rule1 rule2 rule3
           rules-called                    addplus1
   ```

2. ID—Set a descriptive ID name for this session translation.
3. rules-calling—Enter the rules calling in the order in which they should be applied. Multiple rules should be included in quotes and separated by spaces.

```
ACMEPACKET(session-translation)# rules-calling "rule1 rule2 rule3"
```

4. rules-called—Enter the rules called in the order in which they should be applied. Multiple rules should be included in quotes and separated by spaces.

Number Translation Application

To complete your number translation configuration, you must enter into a realm-config or session-agent element and assign session-translations there.

1. Navigate to the chosen element.
   - To move from the session-translation element to the session-agent element, exit out of the session translation element and enter the session agent element.

     ```
     ACMEPACKET(session-translation)# exit
     ACMEPACKET(session-router)# session-agent
     ACMEPACKET(session-agent)#
     ```

   - To move from the session-translation element to the realm-config element, exit from the session translation element to the configuration path.

     ```
     ACMEPACKET(session-translation)# exit
     ACMEPACKET(session-router)# exit
     ```

     Navigate to the ream-config element located in the media-manager path.

     ```
     ACMEPACKET(configure)# media-manager
     ACMEPACKET(media-manager)# realm-config
     ACMEPACKET(realm-config)#
     ```

2. In both realm-config or session agent elements, you must specify an in-translationid and/or an out-translationid in order to configure the number translation.

```
session-agent
        in-translationid
        out-translationid               lrules-out
realm-config
        in-translationid
        out-translationid               lrules-out
```

   Set the following parameters to configure a translation rule:

   - in-translationid—Enter the configured session translation that you want to affect the incoming traffic in this parameter.
   - out-translationid—Enter the configured session translation that you want to affect the outgoing traffic in this parameter.

Other Translations

SIP NAT Translations

There are other translations that occur by way of SIP NAT functionality acting on the SIP R-URI, From-URI, and To URI headers. The translation of URIs in the SIP message occurs as messages are received and sent from the Oracle Enterprise Session Border Controller's SIP proxy. These translations create a bridge between the external and home realms and remove all references to the original IPv4 addressing from the packets sent to the destination network.

The purpose of this translation is to prevent private IPv4 addresses from appearing in SIP message URIs while traveling through the public network. This aspect of the SIP NAT's functionality involves either translating the private address to a public address or encrypting the private address into the URI.

FQDN Mapping

The Oracle Enterprise Session Border Controller maps FQDNs that appear in certain headers of incoming SIP messages to the IPv4 address that the Oracle Enterprise Session Border Controller inserts in outgoing SIP contact headers. The mapped FQDNs are restored in the SIP headers in messages that are sent back to the originator.

This feature is useful to carriers that use IPv4 addresses in the SIP From address to create trunk groups in a PSX for routing purposes. When the carrier's peer uses FQDNs, the carrier is forced to create trunk groups for each possible FQDN that it might receive from a given peer. Similarly, this can apply to SIP Contact and P-asserted-identity headers.

16

Admission Control and QoS

This chapter describes how to configure the Oracle Enterprise Session Border Controller for call admission control and Quality of Service (QoS) monitoring. Call admission control lets you manage call traffic based on several different policies. It is aimed at managing call admission rates in the network, enabling you to maintain suitable QoS levels. A new call is admitted only if it meets the requirements

QoS reporting provides you with real-time evaluation of network and route performance. It lets you contrast internal domain and external domain performance and facilitates SLA verification and traffic engineering.

About Call Admission Control

The Oracle Enterprise Session Border Controller provides call admission control capabilities based on the following policies:

- Bandwidth (single and multi-level policies)
- Session capacity
- Session rate (sustained and burst)

Note: In order to provide admission control for networks to which the Oracle Enterprise Session Border Controller is not directly connected, you need to define multiple realms per network interface.

Bandwidth-Based Admission Control

The Oracle Enterprise Session Border Controller is a policy enforcement point for bandwidth-based call admission control. Sessions are admitted or rejected based on bandwidth policies, configured on the Oracle Enterprise Session Border Controller for each realm.

To manage bandwidth consumption of a network's overall capacity, you can configure aggregate bandwidth policies for each realm.

As the Oracle Enterprise Session Border Controller processes call requests to and from a particular realm, the bandwidth consumed for the call is decremented from the bandwidth pool for that realm. The Oracle Enterprise Session Border Controller determines the required bandwidth from the SDP/H.245 information for SIP and from the OLC sent in the SETUP message for H.323. Any request that would cause the bandwidth constraint to be exceeded is rejected with a SIP 503 Service Unavailable or an H.323 Release Complete.

For example, if an incoming SIP message requests PCMU for a payload/encoding name, a zero (0) payload type, and an 8000 cycle clock rate, the Oracle Enterprise Session Border Controller must determine how much bandwidth is needed.

To accomplish this task, the system checks the media profile values and reserves the bandwidth required for flows. If the required bandwidth for the new flow exceeds the available bandwidth at the time of the request, the Oracle Enterprise Session Border Controller rejects the session.

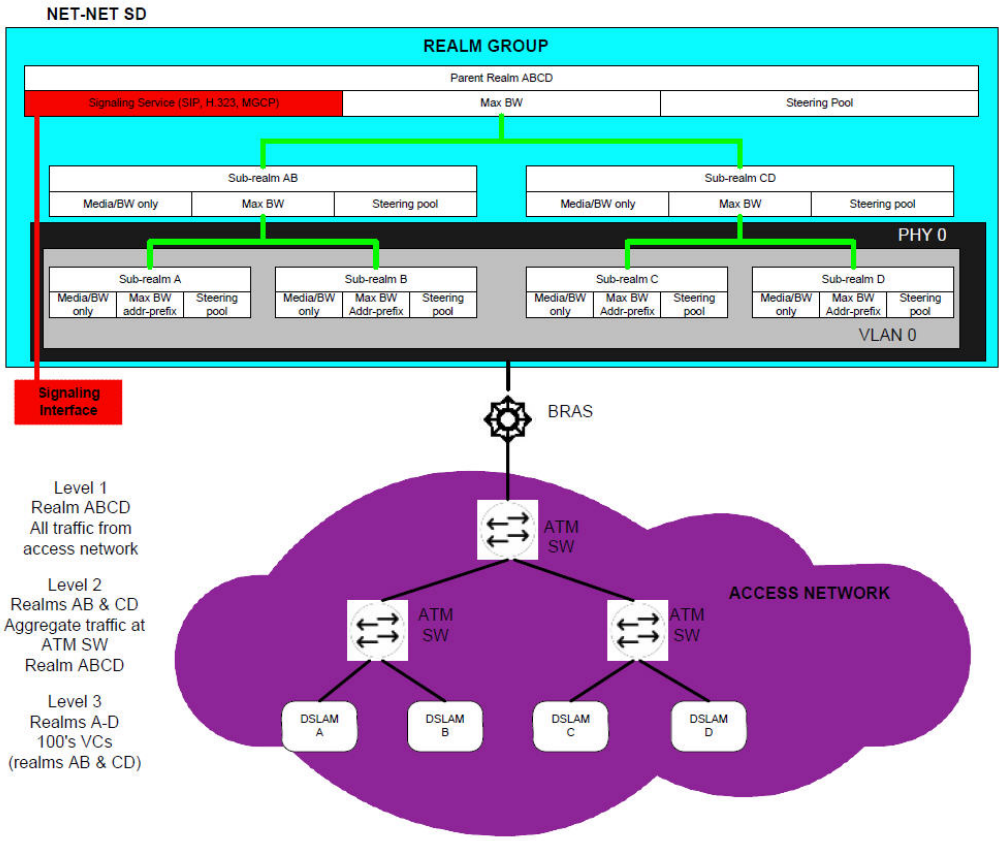
With these mechanisms, the Oracle Enterprise Session Border Controller provides bandwidth-based admission control.

Multi-Level Bandwidth Policy Nesting

Multi-level nesting of bandwidth policy enforcement addresses the following issues:

- Bandwidth over-subscription: access or transit transport networks are aggregated and/or oversubscribed. For example, digital subscriber lines (DSL), Frame Relay (FR), and Asynchronous Transfer Mode (ATM). Admission control policies must reflect access network topology.
- Bandwidth partitioning for multiple services: access or transit bandwidth is partitioned among multiple service profiles (for example, SIP and MGCP) in the same customer network.
- Multi-site VPN environments: admission control must be applied at the site level as well as the VPN level.

The following example illustrates different scenarios; in each there are two or more levels of admission control required. Nested admission control is best depicted by the DSL broadband example.



In DSL access networks, ATM network bandwidth is typically oversubscribed at rates up to 400/1. At Level 3 (above), hundreds of users virtual circuits (VCs) are aggregated to a smaller set of virtual paths (VPs) at each DSLAM. At Level 2, many virtual paths are aggregated at the first ATM switch. Finally, at Level 1, all traffic from all subscribers in the access network is aggregated at the BRAS. Each level of aggregation is oversubscribed, creating the need to perform admission control at each level.

From a Oracle Enterprise Session Border Controller perspective, multiple tiers of realms are supported, each with its unique bandwidth policy. Only the lowest order realm (Level 3) requires an address prefix (that assigned to the DSLAM) that must be used by the Oracle Enterprise Session Border Controller to determine in which realm a user

resides. When a call request to or from a particular user is received, the Oracle Enterprise Session Border Controller checks each realm in the path to determine whether sufficient bandwidth is available to place the call.

Session Capacity- and Rate-based Admission Control

A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. You can define concurrent session capacity and rate attributes for each session agent.

You can configure a set of attributes and constraints for each session agent to support session access control. In this configuration, the Oracle Enterprise Session Border Controller only accepts requests from configured session agents. And you can set up session admission control so that the Oracle Enterprise Session Border Controller limits the number of concurrent inbound and outbound sessions for any known service element.

The Oracle Enterprise Session Border Controller denies a call request to any destination that has exceeded its configured policies for session capacity and session rate. The Oracle Enterprise Session Border Controller might reject the call request back to the originator. If multiple destinations are available, the Oracle Enterprise Session Border Controller will check current capacity and rate for each destination and attempt to route the call only to destinations whose policy limits have not been reached.

You assign a media profile to a session agent and indicate whether the transport protocol is SIP or H.323. If the protocol is H.323, you need to indicate whether the session agent is a gateway or a gatekeeper.

Constraints for Proxy Mode

The Oracle Enterprise Session Border Controller applies session router and session agent constraints when it is in proxy (transaction or stateless) mode if you enable the ACLI constraints parameter for a session agent. However, the Oracle Enterprise Session Border Controller does not track SIP sessions when in transaction mode, so the following session-specific constraints are not applied:

- max-sessions
- max-inbound-sessions
- max-outbound-sessions
- min-seizures
- min-asr

Constraints the Oracle Enterprise Session Border Controller applies are:

- max-burst-rate
- max-inbound-burst-rate
- max-outbound-burst-rate
- max-sustain-rate
- max-inbound-sustain-rate
- max-outbound-sustain-rate

In order to set the desired time windows for computing burst rates and sustain rates, you also need to configure these parameters in the session agent configuration: burst-rate-window and sustain-rate-window. You can also set the time-to-resume and in-service-period parameters to control how long to wait before bringing a session agent back into service after its constraints are no longer exceeded.

CAC Policing and Marking for non-Audio non-Video Media

The Oracle Enterprise Session Border Controller supports non-AVT (audio-visual transport) media profile and media policy configurations.

In previous releases, the Oracle Enterprise Session Border Controller only policed media based on average rate limits configured in media profiles, but these are only applied to AVT. And if there are not required bandwidth or average rate limit values set for the media profile, CAC and policing functions are not applied to media—even if the SDP specifies appropriate bandwidth values. Likewise, ToS markings are not applied for non-AVT media, but only for SIP, H.323, and AVT media types.

With this feature addition, you can now enable your Oracle Enterprise Session Border Controller to handle non-AVT media types like image and text, and use application and data type for policing purposes. Bandwidth CAC support has also been added for non-AVT media types, as has support for the application specific (AS) bandwidth modifier (b=AS:<value>) in the SDP with specification of a defined amount of headroom for that value.

Bandwidth CAC Fallback Based on ICMP Failure

For networks where backup links (operating in active-standby mode) from CE-routers to the MPLS backbone are provisioned with less bandwidth than the primary links, the Oracle Enterprise Session Border Controller can:

- Detect remote link failures
- Trigger bandwidth updates at the realm level when using backup links
- Detect remote link failback to primary

To do so, the Oracle Enterprise Session Border Controller monitors the primary link status using ICMP echo requests (or pings). It issues the pings at regular intervals, forming a heartbeat mechanism. The CE-router can respond to these pings on the primary link, which is represented by the WAN IP address. When this link fails over, the backup link assumes the same WAN IP address but is not responsive to the pings. This way, the Oracle Enterprise Session Border Controller determines failover when the ICMP ping fails.

When there is an ICMP ping failure, the Oracle Enterprise Session Border Controller adjusts the realm's available bandwidth pool from its maximum bandwidth setting to its fallback setting. If the fallback amount is less than the maximum amount, it is possible for the Oracle Enterprise Session Border Controller to start rejecting calls. It does so until enough calls are released to free adequate bandwidth to stay under the fallback limit and still accept calls.

Bandwidth CAC Fallback Based on ICMP Failure Configuration

You can set up ICMP heartbeats and fallback bandwidth pools in the realm configuration. Leaving the icmp-detect-multiplier, icmp-advertisement-interval, or icmp-target-ip parameters blank or set to zero turns the feature off.

To enable bandwidth CAC fallback based on ICMP failure:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. icmp-detect-multiplier—Enter the multiplier you want to use when determining how long to send ICMP pings before considering a target unreachable. This number multiplied by the time you set for the icmp-advertisement-interval determines the length of time. For example, if you set this parameter to 10 and the advertisement interval to 20, the Oracle Enterprise Session Border Controller will send ICMP pings for 200 seconds before declaring the target unreachable.
5. icmp-advertisement-interval—Enter the time in seconds between ICMP pings the Oracle Enterprise Session Border Controller sends to the target. The default is 0.
6. icmp-target-ip—Enter the IP address to which the Oracle Enterprise Session Border Controller should send the ICMP pings so that it can detect when they fail and it needs to switch to the fallback bandwidth for the realm. There is no default.
7. fallback-bandwidth—Enter the amount of bandwidth you want available once the Oracle Enterprise Session Border Controller has determined that the target is unreachable.

If the fallback amount is less than the max-bandwidth value, the Oracle Enterprise Session Border Controller might start to reject calls. It does so until enough calls are released to free adequate bandwidth to stay under the fallback limit and still accept calls.

8. Save and activate your configuration.

Bandwidth CAC for Aggregate Emergency Sessions

You can configure the maximum amount of bandwidth on your Oracle Enterprise Session Border Controller you want used specifically for priority (emergency) calls in the realm configuration's max-priority-bandwidth parameter. You set this limit on a per-realm basis, and the limit is enforced for nested realms. Setting a bandwidth limit specifically for priority calls allows the Oracle Enterprise Session Border Controller to reject calls exceeding the threshold, and also to accept calls that exceed the bandwidth limit for non-priority calls (set in the max-bandwidth parameter).

The bandwidth limit for emergency calls operates in conjunction with the bandwidth limits you can set for all other types of calls. When an emergency call comes in, the Oracle Enterprise Session Border Controller checks the non-priority bandwidth limit. If bandwidth is sufficient, the call goes through and the Oracle Enterprise Session Border Controller decrements the bandwidth used from the pool of the amount available.

However, if a priority call exceeds the max-bandwidth setting, the Oracle Enterprise Session Border Controller checks the max-priority-bandwidth parameter. If is it within the limit for priority calls, the system allows the call and decrements the amount of used bandwidth from what is available.

When there is not enough bandwidth in either the priority or non-priority pool, the Oracle Enterprise Session Border Controller rejects the call with the corresponding error code and reason phrase.

Any bandwidth subtracted from either pool during a session is returned to that pool as soon as the session ends.

Bandwidth CAC for Aggregate Emergency Sessions Configuration

You configure bandwidth CAC for priority calls on a per-realm basis.

Note: This parameter honors the hierarchy of nested realms if you have them configured.

To enable bandwidth CAC for aggregate emergency sessions:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. max-priority-bandwidthEnter the amount of bandwidth you want to want to use for priority (emergency) calls. The system first checks the max-bandwidth parameter, and allows the call if the value you set for priority calls is sufficient. If there is not enough priority and non-priority bandwidth allotted for an incoming call, the Oracle Enterprise Session Border Controller rejects it.

   This parameter defaults to 0. You can enter any value between 0 and 999999999.

5. Save and activate your configuration.

Admission Control for Session Agents

This section explains how to configure session agents for admission control.

Session Agents Admission Control Configuration

To use admission control based on session rate, you need to configure session agent session rate constraints.

To configure session rates:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Enable session agent constraints and then configure the parameters related to session capacity or session rate to set admission control.

   constraints—Enable this parameter. From here you can either configure admission control based on session capacity, session rates, or both. The default value is enabled. The valid values are:

   - enabled | disabled

5. max-sessions—Set the maximum number of sessions (inbound and outbound) allowed by the session agent. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—4294967295

6. max-inbound-sessions—Enter the maximum number of inbound sessions allowed from this session agent. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—999999999

7. max-outbound-sessions—Enter the maximum number of concurrent outbound sessions (outbound from the Oracle Enterprise Session Border Controller) that are allowed from this session agent. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—4294967295

   Note: The number you enter here cannot be larger than the number you entered for max-sessions.

8. max-burst-rate—Enter a number to set how many SIP session invitations or H.323 SETUPs this session agent can send or receive (per second) within the configured burst rate window value. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—4294967295

   For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

   For example, if you enter a value of 50 here and a value of 60 (seconds) for the burst rate window constraint, no more than 300 session invitations can arrive at or leave from the session agent in that 60 second time frame (window). Within that 60-second window, any sessions over the limit of 300 are rejected.

9. max-inbound-burst-rate—Enter the maximum burst rate (number of session invitations per second) for inbound sessions from this session agent. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—999999999

10. max-outbound-burst-rate—Enter the maximum burst rate (number of session invitations per second) for outbound sessions to this session agent.The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

11. max-sustain-rate—Enter a number to set the maximum rate of session invitations (per second) this session agent can send or receive within the current window. The default value is zero (0). The valid range is:

    - Minimum—zero (0)
    - Maximum—4294967295

    The number you enter here must be larger than the number you enter for max-burst-rate.

    For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

    For example, if you enter a value of 50 here and a value of 36 (seconds) for the sustain rate window constraint, no more than 1800 session invitations can arrive at or leave from the session agent in any given 36 second time frame (window). Within that 36 second window, sessions over the 1800 limit are rejected.

12. max-inbound-sustain-rate—Enter the maximum sustain rate (of session invitations allowed within the current window) of inbound sessions from this session agent. This value should be larger than the max-inbound-burst-rate value. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

13. max-outbound-sustain-rate—Enter the maximum sustain rate (of session invitations allowed within the current window) of outbound sessions to this session agent. This value should be larger than the max-outbound-burst-rate value. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

14. burst-rate-window—Enter a number to set the burst window period (in seconds) that is used to measure the burst rate. The term window refers to the period of time over which the burst rate is computed. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—4294967295

15. sustain-rate-window—Enter a number to set the sustained window period (in seconds) that is used to measure the sustained rate. The default value is zero (0), which disables the functionality. The valid range is:

    - Minimum—10
    - Maximum—4294967295

    The value you set here must be higher than or equal to the value you set for the burst rate window.

    Note: If you are going to use this parameter, you must set it to a minimum value of 10.

The following example shows session agent constraints that are enabled and the session capacity parameters have been configured. Other session agent parameters have been omitted for brevity.

```
session-agent
constraints                    enabled
max-sessions                   355
max-inbound-sessions          355
max-outbound-sessions          355
```

The following example shows session agent constraints are enabled and the session rate parameters have been configured. Other session agent parameters have been omitted for brevity.

```
session-agent
max-burst-rate                 0
```

```
max-inbound-burst-rate          10
max-outbound-burst-rate         1
max-sustain-rate                3000
max-inbound-sustain-rate        0
max-outbound-sustain-rate       0
burst-rate-window               0
sustain-rate-window             0
```

Realm Bandwidth Configuration

To configure admission control based on bandwidth, you set the max and min bandwidth parameters in the realm configuration.

To configure realm bandwidth:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. Configure the maximum bandwidth.

   max-bandwidth—Enter a number that sets the maximum bandwidth for dynamic flows to/from the realm in kilobits (Kbps) per second. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—4294967295

   The following example shows the maximum bandwidth for the realm has been configured. All other realm parameters have been omitted for brevity.

   ```
   realm-config
   max-bandwidth                   64000
   ```

SIP Admission Control Configuration

You can configure the registered endpoint to accept and process requests from SIP realms. If a request does not meet the criteria of the option you choose here, it is rejected with a 403 (Forbidden) response.

To configure admission control:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. Type sip-ports and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(sip-interface)# sip-port
   ACMEPACKET(sip-port)#
   ```

5. Set the criteria for admission control.

allow-anonymous—Enter the anonymous connection mode you want applied when SIP requests are processed. The default value is all.

The following are valid values:

- all—No ACL is applied and all anonymous connections are allowed.
- agents-only—Only requests from configured session agents are processed. The Oracle Enterprise Session Border Controller responds to all other requests with a forbidden response.
- realm-prefix—Only requests from session agents and addresses matching the realm's address prefix are processed. All other requests are rejected with a 403 (Forbidden) response.
- registered—Only requests from session agents and registered endpoints are processed. REGISTER allowed from any endpoint.
- registered-prefix—Only requests from session agent and registered endpoint addresses that match the realm's realm prefix are processed.

  The following example shows the allow-anonymous parameter that has been configured to allow only requests from session agents and registered endpoints. All other session agent parameters following the allow-anonymous parameters are omitted for brevity.

```
sip-port
                address
                port                            5060
                transport-protocol              UDP
                allow-anonymous                 registered
```

H.323 Admission Control Configuration

You can configure the endpoint to allow accept and process requests from a H.323 realm. If a request does not meet the criteria you set here, it is rejected.

To configure admission control:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type h323 and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. Type h323-stacks and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
```

5. Set the criteria upon which you want to base admission control.

   allow-anonymous—Enter the anonymous connection option (mode) you want applied to the processing of H.323 requests. The default value is all.

   The following are valid values:

   - all—No ACL is applied and all anonymous connections are allowed.
   - agents-only—Only requests from configured session agents are processed.
   - realm-prefix—Only requests from session agents and addresses matching the realm's address prefix are processed. All other requests are rejected.

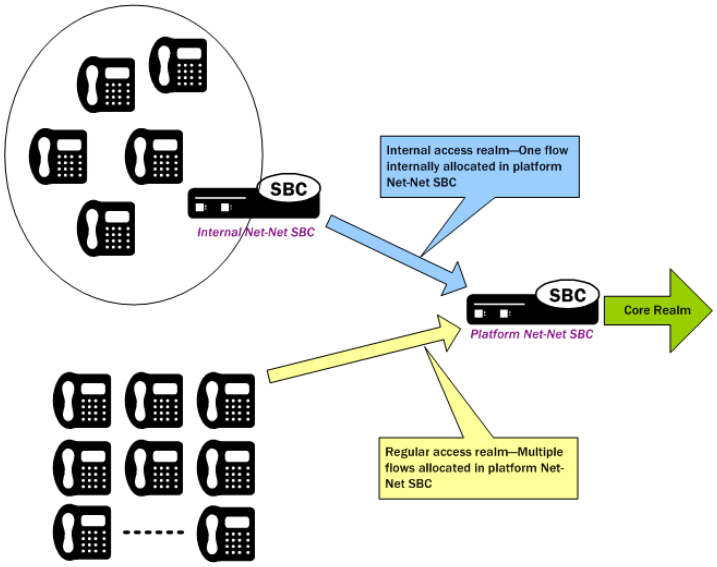
     The following example shows the allow-anonymous parameter has been configured to allow only requests from configured session agents. All other h.323-stack parameters are omitted for brevity.

```
h323-stack
allow-anonymous                 agents-only
```

Session Agent Minimum Reserved Bandwidth

You can assign session agents minimum bandwidth, applicable in accessOracle Enterprise Session Border Controller deployments. Assigning a session agent minimum bandwidth can prevent overloading other network devices—such as another Oracle Enterprise Session Border Controller configured as a session agent. Doing so assures signaling bandwidth and availability to the endpoints behind this Oracle Enterprise Session Border Controller. This feature is only available on the Acme Packet 3820 and Acme Packet 4500.

In the following diagram, the internal Oracle Enterprise Session Border Controller is configured as a session agent on the platform Oracle Enterprise Session Border Controller (which conveys traffic to the core realm). Setting up bandwidth reservation allows for the creation of only one allocated flow, and secures bandwidth for all the SIP clients behind the internal Oracle Enterprise Session Border Controller. Contrast this scenario with the one where the platform Oracle Enterprise Session Border Controller must allocate multiple flows for many SIP clients.



When you configure minimum reserved bandwidth for session agent to a non-zero value, the Oracle Enterprise Session Border Controller allocates a separate pipe for per session agent. This is achieved by setting up an access control configuration in a specific way, instructing the Oracle Enterprise Session Border Controller to use a minimum number of transmission timeslots the individual pipe is guaranteed to receive.

This feature works across all signaling services: SIP, H.323, and MGCP. No more than 4000 session pipes are supported.

Session Agent Minimum Reserved Bandwidth Configuration

For the feature to work, you must set up an access control configuration with the settings required in the instructions and examples below.

To configure minimum reserved bandwidth for session agents:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type access-control and press Enter.

```
ACMEPACKET(session-router)# access-control
ACMEPACKET(access-control)#
```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. realm-id—Enter the name of a valid realm.
5. application-protocol—Enter a valid application protocol. There is no default for this parameter, and valid values are: SIP, H.323, or MGCP.
6. access—Set this parameter to permit (default).
7. trust-level—Set this parameter to high, changing it from the default (none).
8. minimum-reserved-bandwidth—Enter the minimum reserved bandwidth you want for the session agent, and that will trigger the creation of a separate pipe for it. Only a non-zero value will allow the feature to work properly, along with the other required values set out in these instructions. The default is 0, and the maximum is 0xffffffff (or 4294967295).
9. Save and activate your configuration.

Aggregate Session Constraints for SIP

You can set a full suite of session constraints and then apply them to a SIP interface. The session constraints configuration contains many of the same parameters as the session agent, so you can configure a group of constraints and then apply them to a SIP interface/

The SIP interface configuration's constraint-name parameter invokes the session constraint configuration you want to apply. Using the constraints you have set up, the Oracle Enterprise Session Border Controller checks and limits traffic according to those settings for the SIP interface. Of course, if you do not set up the session constraints or you do not apply them in the SIP interface, then that SIP interface will be unconstrained. If you apply a single session-constraint element to multiple SIP interfaces, each SIP interface will maintain its own copy of the session-constraint.

SIP interfaces now have two states: "In Service" and "Constraints Exceeded." When any one of the constraints is exceeded, the status of the SIP interface changes to Constraints Exceeded and remains in that state until the time-to-resume period ends. The session constraint timers that apply to the SIP interface are the time-to-resume, burst window, and sustain window.

Aggregate Session Constraints Configuration

This section shows you how to configure aggregate session constraints and then apply them to a SIP interface.

The session constraints configuration contains many of the same parameters as the session agent does; it also incorporates the changes to the session agent parameters that are described in this section.

To configure the session constraints:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type session-constraints and press Enter.

```
ACMEPACKET(session-router)# session-constraints
```

4. name—Enter the name for this session constraints configuration; this is a unique identifier that you will use in the SIP interface when you want the session constraints applied there. This is a required parameter that has no default.
5. state—Enable this parameter to use these session constraints. The default value is enabled. The valid values are:
   - enabled | disabled
6. max-sessions—Enter the maximum sessions allowed for this constraint. The default value is zero (0). The valid range is:

- Minimum—0
- Maximum—999999999

7. max-outbound-sessions—Enter the maximum outbound sessions allowed for this constraint. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—999999999

8. max-inbound-sessions—Enter the maximum inbound sessions allowed for this constraint. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—999999999

9. max-burst-rate—Enter the maximum burst rate (invites per second) allowed for this constraint. This value should be the sum of the max-inbound-burst-rate and the max-outbound-burst-rate. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—999999999

10. max-sustain-rate—Enter the maximum rate of session invitations per second allowed within the current window for this constraint. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

      For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

11. max-inbound-burst-rate——Enter the maximum inbound burst rate (number of session invitations per second) for this constraint. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

12. max-inbound-sustain-rate—Enter the maximum inbound sustain rate (of session invitations allowed within the current window) for this constraint. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

      For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

13. max-outbound-burst-rate——Enter the maximum outbound burst rate (number of session invitations per second) for this constraint. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

14. max-outbound-sustain-rate—Enter the maximum outbound sustain rate (of session invitations allowed within the current window) for this constraint. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

      For the sustained rate, the Oracle Enterprise Session Border Controller maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

15. time-to-resume—Enter the number of seconds after which the SA (Session Agent) is put back in service (after the SA is taken OOS (Out Of Service) because it exceeded some constraint). The default value is zero (0). The valid range is:

- Minimum—0
- Maximum—999999999

16. ttr-no-response—Enter the time delay in seconds to wait before the SA (Session Agent) is put back in service (after the SA is taken OOS (Out Of Service) because it did not respond to the Oracle Enterprise Session Border Controller). The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

17. in-service-period—Enter the time in seconds that elapses before an element (like a session agent) can return to active service after being placed in the standby state. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

18. burst-rate-window—Enter the time in seconds that you want to use to measure the burst rate; the window is the time over which the burst rate is calculated, and is used for the over all burst rate as well as the inbound and outbound burst rates. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

19. sustain-rate-window—Enter the time in seconds used to measure the sustained rate; the window is the time over which the sustained rate is calculated, and is used for the over all sustained rate as well as the inbound and outbound sustained rates. The default value is zero (0). The valid range is:

    - Minimum—0
    - Maximum—999999999

Applying Session Constraints in a SIP Interfaces

In the SIP interface, there is a new parameter that allows you to use a set of session constraints for that interface; the parameter is called constraint-name.

To apply session constraints to a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
```

4. constraint-name—Enter the name of the session constraints configuration that you want to apply to this SIP interface. There is no default for this parameter.
5. Save and activate your configuration.

Configuring CAC Policing and Marking for non-Audio non-Video Media

In the media profile and the media policy configurations, the following values have been added for the media-type parameter:

- application | data | image | text

For the media policy, these new values apply to ToS marking.

Support for the AS Bandwidth Modifier

Two new parameters have been added to the media profile configuration:

- sdp-bandwidth—Enable or disable the use of the AS modifier in the SDP if the req-bandwidth and sdp-rate-limit-headroom parameters are not set to valid values in a corresponding media profile. The default value is disabled. The valid values are:
  - enabled | disabled
- sdp-rate-limit-headroom—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the average-rate-limit (rate limit for the RTP flow). The default value is zero (0). The valid range is:
  - Minimum—0
  - Maximum—100

The following conditions apply to the use and application of these two new parameters:

- If the amount of required bandwidth is not specified in the media profile (req-bandwidth) for the media type in the m= line of the SDP, then the value specified in the AS modifier is used. The Oracle Enterprise Session Border Controller only uses the AS value if you set the new sdp-bandwidth to enabled.
- If the average rate limit value for RTP flows is not specified in the media profile (average-rate-limit) for the media type in the m= line of the SDP, then the value specified in the AS modifier is used. The system only uses the AS value if you set the new sdp-bandwidth to enabled. When calculating the average rate rate limit that it will use based on the AS modifier, the Oracle Enterprise Session Border Controller applies the percentage set in the sdp-rate-limit-headroom parameter.
- The Oracle Enterprise Session Border Controller uses the value specified in the AS modifier (if sdp-bandwidth is enabled, and req-bandwidth is set to 0) along with the user-cac-bandwidth value set in the realm configuration; this works the same way that the req-bandwidth parameter does.
- The system uses the value specified in the AS modifier (if sdp-bandwidth is enabled, and req-bandwidth is set to 0) along with the max-bandwidth value set in the realm configuration; this works the same way that the req-bandwidth parameter does.

Media Profile Configuration

To set any of the new media types in the media profile configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type media-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. media-type—Enter the media type that you want to use for this media profile. The valid values are:
   - audio | video | application | data | image | text
5. Save and activate your configuration.

To set any of the new media types in the media policy configuration:

6. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

7. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

8. Type media-policy and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-policy
ACMEPACKET(media-policy)#
```

9. media-type—Enter the media type that you want to use for this media profile. The valid values are:
   - audio | video | application | data | image | text
10. Save and activate your configuration.

AS Modifier and Headroom Configuration

To enable AS modifier use and establish the percentage of headroom to use:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type media-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. sdp-bandwidth—Enable this parameter to use the AS bandwidth modifier in the SDP. The default is disabled. Valid values are:
   - enabled | disabled
5. sdp-rate-limit-headroom—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the average-rate-limit (rate limit for the RTP flow). The default is 0. The valid range is:
   - Minimum—0
   - Maximum—100
6. Save and activate your configuration.

Offerless Bandwidth CAC for SIP

For SIP sessions offerless INVITEs (i.e., INVITEs that have no SDP offer), the Oracle Enterprise Session Border Controller can reserved bandwidth and support the session if you set up applicable media profile associations in the global SIP configuration. Otherwise, the Oracle Enterprise Session Border Controller terminates these sessions.

You configure support for offerless bandwidth CAC by setting up your global SIP configuration with the options parameters set to offerless-media-bw-profiles. The option takes multiple media profile names as values to apply when treating offerless INVITEs. When such an INVITE arrives and your configuration supports this option, the Oracle Enterprise Session Border Controller checks and reserves bandwidth for the session. If there is insufficient bandwidth to reserve, the Oracle Enterprise Session Border Controller terminates the session. Otherwise, the actual SDP negotiation takes place unaffected while the Oracle Enterprise Session Border Controller forwards the offerless INVITE. Once the negotiation completes, the Oracle Enterprise Session Border Controller updates bandwidth reservation.

If the called party's actual bandwidth needs exceed available bandwidth, the Oracle Enterprise Session Border Controller must terminate the session, even if the session is ringing or answered. To minimize this occurrence as much as possible, you should consider all case scenarios when you select media profiles to use with the offerless-media-bw-profiles option.

Offerless Bandwidth CAC for SIP Configuration

To configure offerless bandwidth CAC for SIP:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-config and press Enter. If you are editing a pre-existing configuration, you needs to select it before you can make changes.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. options——Your entry will look like this:

```
ACMEPACKET(sip-config)# options offerless-bw-media-profiles=PCMU,G729
```

   You can you the plus sign (+) and the minus sign (-) to add and remove values from the options list.

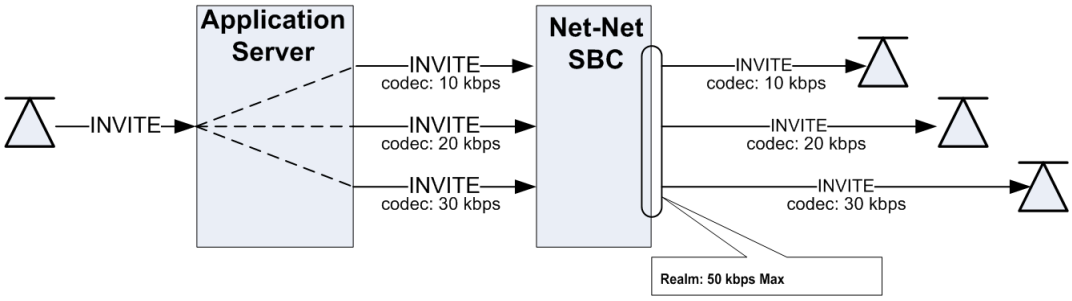
5. Type done and continue

Shared CAC for SIP Forked Calls

A forked call is one which has multiple INVITEs for the same call. For example, if an Application Server in the provider core network forks a call attempt, the application server sends several INVITEs for the same call toward the Oracle Enterprise Session Border Controller. Each INVITE is destined for a unique device that belongs to the same user. Ideally, that user will only answer one device. The Oracle Enterprise Session Border Controller treats each INVITE as a unique call request.

By default, each of the multiple INVITE forks are checked against CAC bandwidth limits, and thus they each consume bandwidth resources when they are received, even though only one of the forks will succeed in establishing a permanent session. Therefore, for many operators the CAC behavior of the SD is too restrictive and results in rejected call attempts which should have been allowed.

The following diagram shows a forked call scenario. The total bandwidth counted against the realm is 60 kbps. If the realm has a bandwidth ceiling of 50 kbps, one of the INVITEs will be rejected.



You can, however, enable the system to enforce CAC limits only once for SIP forked calls as long as the calls are identified as such, meaning that they will use the same bandwidth resources. The Oracle Enterprise Session Border Controller counts the forked call's most bandwidth-hungry codec at the time it arrives at the Oracle Enterprise Session Border Controller. In the above diagram, with shared bandwidth for forked calls enabled, the Oracle Enterprise Session Border Controller counts 30 kbps against the realm's total bandwidth after that INVITE arrives, even after the first two INVITES have passed into the final realm.

Bandwidth Sharing Scenarios

The following table summarizes how bandwidth would be shared given certain ingress and egress realms with this feature enabled. Realms A and C are call ingress realms.; realms B and D are egress realms. For the bandwidth to be shared, Call A and Call B must have the same forked Call-ID in the P-Multiring-Correlator header and be entering or exiting the Oracle Enterprise Session Border Controller on the same realm.

CALL A					
CALL B		Ingress Realm A	Egress Realm B	Ingress Realm C	Egress Realm D
	Ingress Realm A	bandwidth shared	N/A	bandwidth not shared	N/A
	Egress Realm B	N/A	bandwidth shared	N/A	bandwidth not shared
	Ingress Realm C	bandwidth not shared	N/A	bandwidth shared	N/A
	Egress Realm D	N/A	bandwidth not shared	N/A	bandwidth shared

Bandwidth Sharing Configuration

To enable bandwidth sharing of forked calls, set the forked-cac-bw parameter in the SIP profile configuration to shared. Although there are other parameters available in the SIP profile configuration, you only have to set the name and the forked-cac-bw values to use this feature.

After you set up the SIP profile, you apply it to a realm, SIP interface, or session agent.

Configuring a SIP Profile

The SIP profile is an element in the ACLI's session-router path, and you can configure multiple SIP profiles.

To configure a SIP profile:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-profile and press Enter.

```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```

4. name—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's name when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. forked-cac-bw—Set this parameter to shared if you want forked sessions to share bandwidth resources, or set it to per-session if you want bandwidth to be counted for each session individually. There is no default for this parameter, and leaving it blank means:
   - For an ingress session agent without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Oracle Enterprise Session Border Controller will reference the associated realm.
   - For an ingress realm without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Oracle Enterprise Session Border Controller will reference the associated SIP interface.
   - For an ingress SIP interface without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Oracle Enterprise Session Border Controller will not perform bandwidth sharing for forked calls.
6. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- realm-config
- sip-interface

- session-agent

To apply a SIP profile to a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. sip-profile—Enter the name of SIP profile configuration that includes the forked-cac-bandwidth parameter configured.
5. Save your work.

RADIUS Accounting Support

VSA 171, Acme-Session-Forked-Call-Id, is part of the Oracle RADIUS dictionary. The VSA is a string value, and appears as the header-value without the header parameters from the P-Multiring-Correlator header for a session identified as part of a forked call.

Monitoring

Using the ACLI show sipd forked command, you can display the total number of forked sessions the Oracle Enterprise Session Border Controller received and the total number it rejected. The Oracle Enterprise Session Border Controller counts forked sessions when it receives a dialog-creating INVITE and is enabled to shared bandwidth. Further, it counts as forked all session with the P-Multiring-Correlator header.

```
ACMEPACKET# show sipd forked
11:19:20-116
Forked Sessions               ---- Lifetime ----
                       Recent      Total  PerMax
Forked Sessions             0          0       0
Forked Sessions Rej         0          0       0
```

Conditional Bandwidth CAC for Media Release

The Oracle Enterprise Session Border Controller supports conditional call admission control (CAC) using the SIP profile configuration. With this feature enabled, you can allow the conditional admission of SIP calls that could potentially have their media released instead of risking the possible rejection of those calls due to internal bandwidth limits.

About Conditional Bandwidth CAC for Media Release

The Oracle Enterprise Session Border Controller performs bandwidth CAC for SIP per realm, for each Address of Record (AoR) or IP address. The system checks bandwidth limits based on the codecs listed in SDP. If a new SIP INVITE contains codecs in an SDP message that exceed bandwidth available for a given resource, the system rejects that INVITE. This check occurs both on the ingress and egress sides of a call, and both sides must have enough available resources to support the call for it to be admitted.

In the case of calls where media is released, the Oracle Enterprise Session Border Controller does not count bandwidth consumed by the call. However, this exemption is not given until the media is actually released—and media release conditions are unknown at the time SIP INVITE is admitted. This is because an INVITE received on one side of the Oracle Enterprise Session Border Controller is only media-released when that INVITE is routed back through the Oracle Enterprise Session Border Controller as a hairpin or other multi-system media release. So there

has to be enough bandwidth for the initial INVITE; otherwise, and even if the INVITE is a candidate for media release, it will be rejected.

When there is a significant volume of such calls—ones that are candidates for media release, but cannot be admitted because of CAC limits—it becomes important to admit them so long as they truly end in media release. This feature thus allows admission of SIP calls that might otherwise be rejected because of bandwidth limitations when the far-end of the call causes media to be released.

Details and Conditions

This feature applies in a two system scenario. In order to track a call as a candidate for provisional media release, the access-side Oracle Enterprise Session Border Controller adds a Require: header with an option tag to the INVITE or UPDATE message on egress. The option tag is configurable in the sip config option. The default is com.acmepacket.cac .

The following sections describe when the SIP INVITE or SIP UPDATE are:

- initially received by the Oracle Enterprise Session Border Controller
- received by the second Oracle Enterprise Session Border Controller

INVITEs UPDATEs Initially Received By Oracle Enterprise Session Border Controller

When the Oracle Enterprise Session Border Controller first receives an INVITE or UPDATE message, it considers if it should be admitted provisionally or rejected outright due to CAC bandwidth constraints. If the INVITE or UPDATE is admitted provisionally, a Require: header is inserted on egress from the system.

The Oracle Enterprise Session Border Controller inserts the Require header on egress under these conditions:

- It receives an INVITE / UPDATE with no or a non-matching Require header.
- The egress conditional cac admit parameter in the SIP profile on the egress realm, SIP interface, session agent is set to enabled in the egress realm
- The request would otherwise be rejected because of current bandwidth CAC limits in the ingress OR egress realms
- The call is a candidate for media-release in the ingress realm

A call is considered a candidate for media-release when the ingress realm has any of these parameters set to disabled:

- mm-in-realm
- mm-in-network
- mm-same-ip
- mm-in-system

INVITEs UPDATEs Received by Second SBC

The second Oracle Enterprise Session Border Controller receives the INVITE or UPDATE with the newly inserted Require: header. Standard SIP convention indicates that if the UAS receiving the request does not know how to handle the Require header, the request should be rejected.

When the following three conditions are met, the INVITE is permitted into the system for processing:

- The ingress conditional cac admit in the SIP profile on the ingress realm, SIP interface, session agent parameter is set to enabled
- The con-cac-tag sip config option is configured to the same value as the received Require header’s option tag
- The call is a candidate for media-release

The call is considered a candidate for media-release on the second system (indicated by the ingress conditional cac admit parameter is set to enabled) when either the ingress or egress realms have any of these parameters set to disabled:

- mm-in-realm
- mm-in-network
- mm-same-ip

- mm-in-system

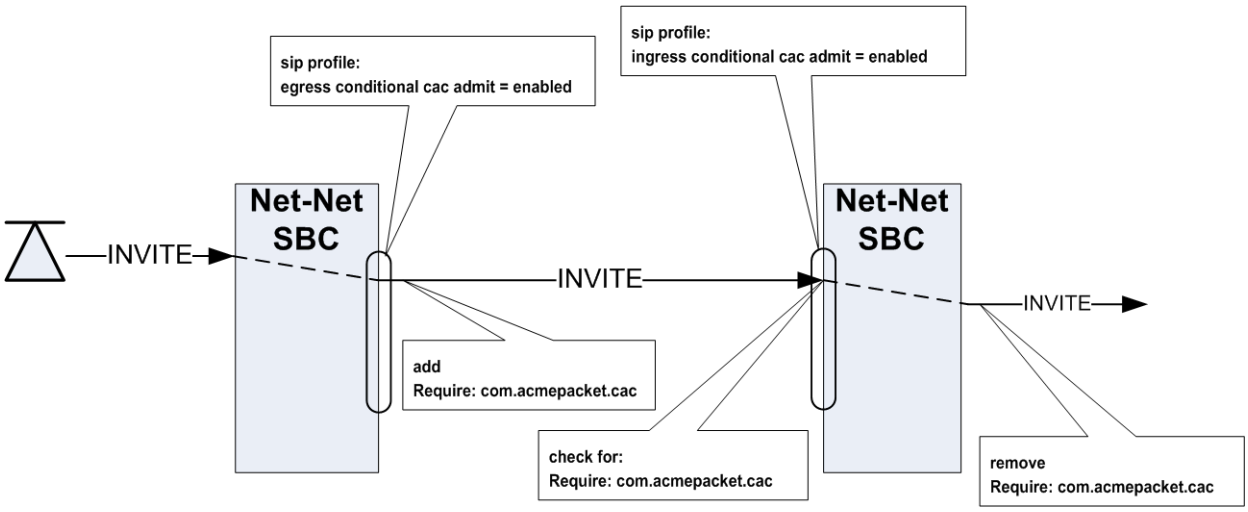
and the following parameter is set to enabled:

- msm-release

If the call, as received by the second system is not considered a candidate for release, the INVITE or UPDATE is failed with a 503 Insufficient Bandwidth message.

After the INVITE has been processed by the Oracle Enterprise Session Border Controller, the Require: header is removed upon egress from the system.

The following diagram shows the two-system scenario:



Conditional Admission with Per-user CAC

In the event that the per-user CAC feature is also being used, and per-user CAC bandwidth is exceeded, the Oracle Enterprise Session Border Controller also uses this option tag mechanism. However, if the per-user CAC implementation does count bandwidth regardless of media-release, then the Oracle Enterprise Session Border Controller will reject calls exceeding the per-user CAC limits when it receives them.

On the second system, when the per-user CAC feature is being used, the Oracle Enterprise Session Border Controller will perform the same option tag mechanism based on if the ingress conditional cac admit parameter is enabled.

Conditional Bandwidth CAC Configuration

You enable this feature by first configuring a SIP profile, and then applying the profile to any of these:

- realm
- SIP interface
- SIP session agent

SIP Profile Configuration

The SIP profile is an element in the ACLI's session-router path, and you can configure multiple SIP profiles. Though this configuration contains additional parameters, you do not have to use them for the conditional bandwidth CAC for media release.

To configure a SIP profile:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type account-config and press Enter.

```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```

4. name—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's name when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. ingress-conditional-cac-admit—Set this parameter to enabled to process an INVITE with a Require tag as received on an ingress interface. You can set this parameter to disabled if you do not want to use this feature on the ingress side. There is no default for this parameter.
6. egress-conditional-cac-admit—Set this parameter to enabled if you want to use conditional bandwidth CAC for media release for calls that are first received by this system. This results in option tags being inserted on the INVITE's egress if the conditional CAC conditions are met. You can set this parameter to disabled if you do not want to use this feature. There is no default for this parameter.
7. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- realm-config
- sip-interface
- session-agent

To apply a SIP profile to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. sip-profile—Enter the name of SIP profile configuration you want to use for conditional bandwidth CAC for media release for this SIP interface. This value is blank by default, but it must be the value of the name parameter from a valid SIP profile.
5. Save your work.

Configuring Require Header Option Tag

You may change the Require: header's option tag from the default com.acmepacket.cac to one of your own choosing. Remember that both systems' option tags must match exactly.

To configure the Require: header's option tag:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type sip-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
```

4. Use the ACLI select command so that you can work with the SIP configuration.

```
ACMEPACKET(sip-config)# select
```

5. options—Set the options parameter by typing +options, a Space, the option name con-cac-tag= your-new-tag, and then press Enter.

```
ACMEPACKET(sip-config)# options +con-cac-tag=com.test.cac
```

6. Save your work.

About QoS Reporting

This section describes the Oracle Enterprise Session Border Controller QoS reporting. QoS reporting provides you with real-time evaluation of network and route performance. It lets you contrast internal domain and external domain performance and facilitates SLA verification and traffic engineering. Oracle Enterprise Session Border Controller QoS reporting is a measurement tool that collects statistics on Voice over IP (VoIP) call flows for SIP and H.323. To provide information, the Oracle Enterprise Session Border Controller writes additional parameters to the Remote Authentication Dial-in User Service (RADIUS) call record. To provide information, the Oracle Enterprise Session Border Controller writes additional parameters to the Remote Authentication Dial-in User Service (RADIUS) call record and Historical Data Recording (HDR) records.

You can use QoS statistics for SLA customer reporting, fault isolation, SLA verification, and traffic analysis. The Oracle Enterprise Session Border Controller employs specialized hardware to inspect Real-Time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP) flows while maintaining wire-speed packet forwarding. QoS metrics are collected and reported on a per-session and per call-leg basis. These metrics are reported through real-time RADIUS records along with call accounting data.

Overview

When a conversation is established between two endpoints, two flows are present in each direction:

- RTP flow carries traffic between endpoints with a predictable packet arrival rate. The packets headers have sequence numbers that are used to determine whether packets are missing or lost.
- RTCP flow carries information about the RTP flow and keeps a different record. The RTCP packets contain timestamps based on Network Time Protocol (NTP).

QoS Statistics

Reported QoS data includes the following per-flow statistics:

- RTP and RTCP lost packets—Count of lost packets for both RTP and RTCP based on comparing the sequence numbers since the beginning of the call or the last context memory poll.
- RTP and RTCP average jitter—Incremental number of packets for both RTP and RTCP that have been used to generate the total and max jitter since the beginning of the call or the last context memory poll. The incremental accumulated jitter (in milliseconds) over all the packets received.
- RTP and RTCP maximum jitter—Maximum single jitter value (in milliseconds) for both RTP and RTCP from all the packets since the beginning of the call or the last context memory poll.
- RTCP average latency—Number of RTCP frames over which latency statistics have been accumulated and the incremental total of latency values reported since the beginning of the call or the last context memory poll.
- RTCP maximum latency—Highest latency value measured since the beginning of the call or the last context memory poll.
- RTP packet count
- RTP bytes sent and received
- RTCP lost packets—RTP lost packets reported in RTCP packets.
- ATP lost packets—Lost packets determined by monitoring RTP sequence numbers.

- R-Factor and MOS data—R-Factor and MOS data for the calling and called segments at the end of a session

RADIUS Support

All the QoS statistics go into the RADIUS CDR. If a RADIUS client is configured on the Oracle Enterprise Session Border Controller, any time a call occurs a record is generated and sent. Only Stop RADIUS records contain the QoS statistic information.

Only RADIUS Stop records contain QoS information. For non-QoS calls, the attributes appear in the record, but their values are always be zero (0). When you review the list of QoS VSAs, please note that “calling” in the attribute name means the information is sent by the calling party and called in the attribute name means the information is sent by the called party.

The following example shows a CDR that includes QoS data:

```
Wed Jun 13 18:26:42 2007
        Acct-Status-Type = Stop
        NAS-IP-Address = 127.0.0.100
        NAS-Port = 5060
        Acct-Session-Id = "SDgtu4401-c587a3aba59dcae68ec76cb5e2c6fe6f-v3000i1"
        Acme-Session-Ingress-CallId =
"8EDDDC21D3EC4A218FF41982146844310xac1ec85d"
        Acme-Session-Egress-CallId = "SDgtu4401-
c587a3aba59dcae68ec76cb5e2c6fe6f-v3000i1"
        Acme-Session-Protocol-Type = "SIP"
        Calling-Station-Id = ""9998776565" <sip:
9998776565@10.10.170.2:5060>;tag=2ed75b8317f"
        Called-Station-Id = "<sip:7143221099@10.10.170.2:5060>"
        Acct-Terminate-Cause = User-Request
        Acct-Session-Time = 7
        h323-setup-time = "18:24:36.966 UTC JUN 13 2007"
        h323-connect-time = "18:24:37.483 UTC JUN 13 2007"
        h323-disconnect-time = "18:24:44.818 UTC JUN 13 2007"
        h323-disconnect-cause = "1"
        Acme-Session-Egress-Realm = "peer"
        Acme-Session-Ingress-Realm = "core"
        Acme-FlowID_FS1_F = "localhost:65544"
        Acme-FlowType_FS1_F = "PCMA"
        Acme-Flow-In-Realm_FS1_F = "core"
        Acme-Flow-In-Src-Addr_FS1_F = 10.10.170.15
        Acme-Flow-In-Src-Port_FS1_F = 49156
        Acme-Flow-In-Dst-Addr_FS1_F = 10.10.170.2
        Acme-Flow-In-Dst-Port_FS1_F = 31008
        Acme-Flow-Out-Realm_FS1_F = "peer"
        Acme-Flow-Out-Src-Addr_FS1_F = 10.10.130.2
        Acme-Flow-Out-Src-Port_FS1_F = 21008
        Acme-Flow-Out-Dst-Addr_FS1_F = 10.10.130.15
        Acme-Flow-Out-Dst-Port_FS1_F = 5062
        Acme-Calling-RTCP-Packets-Lost_FS1 = 0
        Acme-Calling-RTCP-Avg-Jitter_FS1 = 15
        Acme-Calling-RTCP-Avg-Latency_FS1 = 0
        Acme-Calling-RTCP-MaxJitter_FS1 = 15
        Acme-Calling-RTCP-MaxLatency_FS1 = 0
        Acme-Calling-RTP-Packets-Lost_FS1 = 0
        Acme-Calling-RTP-Avg-Jitter_FS1 = 3
        Acme-Calling-RTP-MaxJitter_FS1 = 44
        Acme-Calling-Octets_FS1 = 957
        Acme-Calling-Packets_FS1 = 11
        Acme-FlowID_FS1_R = "localhost:65545"
        Acme-FlowType_FS1_R = "PCMA"
        Acme-Flow-In-Realm_FS1_R = "peer"
        Acme-Flow-In-Src-Addr_FS1_R = 10.10.130.15
        Acme-Flow-In-Src-Port_FS1_R = 5062
        Acme-Flow-In-Dst-Addr_FS1_R = 10.10.130.2
```

```
        Acme-Flow-In-Dst-Port_FS1_R = 21008
        Acme-Flow-Out-Realm_FS1_R = "core"
        Acme-Flow-Out-Src-Addr_FS1_R = 10.10.170.2
        Acme-Flow-Out-Src-Port_FS1_R = 31008
        Acme-Flow-Out-Dst-Addr_FS1_R = 10.10.170.15
        Acme-Flow-Out-Dst-Port_FS1_R = 49156
        Acme-Called-RTCP-Packets-Lost_FS1 = 0
        Acme-Called-RTCP-Avg-Jitter_FS1 = 13
        Acme-Called-RTCP-Avg-Latency_FS1 = 0
        Acme-Called-RTCP-MaxJitter_FS1 = 21
        Acme-Called-RTCP-MaxLatency_FS1 = 0
        Acme-Called-RTP-Packets-Lost_FS1 = 0
        Acme-Called-RTP-Avg-Jitter_FS1 = 0
        Acme-Called-RTP-MaxJitter_FS1 = 3
        Acme-Called-Octets_FS1 = 77892
        Acme-Called-Packets_FS1 = 361
        Acme-Firmware-Version = "C5.0.0"
        Acme-Local-Time-Zone = "Time Zone Not Set"
        Acme-Post-Dial-Delay = 110
        Acme-Primary-Routing-Number = "sip:7143221099@10.10.170.2:5060"
        Acme-Ingress-Local-Addr = "10.10.170.2:5060"
        Acme-Ingress-Remote-Addr = "10.10.170.15:5060"
        Acme-Egress-Local-Addr = "10.10.130.2:5060"
        Acme-Egress-Remote-Addr = "10.10.130.15:5060"
        Acme-Session-Disposition = 3
        Acme-Disconnect-Initiator = 2
        Acme-Disconnect-Cause = 16
        Acme-SIP-Status = 200
        Acme-Egress-Final-Routing-Number = "sip:7143221099@10.10.130.15:5060"
        Acme-CDR-Sequence-Number = 14
        Client-IP-Address = 172.30.20.150
        Acct-Unique-Session-Id = "0832b03cd3a290b3"
        Timestamp = 1181773602
```

Configuring QoS

This section explains how to configure QoS. To generate QoS metrics, you need to enable QoS for the realm of the originating caller. The ingress realm determines whether QoS is turned on for a specific flow.

Note: If you run with QoS turned on one side only and disabled on the other you lose the ability to measure latency through the use of RTCP timestamps.

QoS Configuration

To enable QoS:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

4. qos-enable—Enable this parameter. The default value is disabled.

Accounting Configuration for QoS

This section explains how to configure the account configuration and account servers so you can use the Oracle Enterprise Session Border Controller in conjunction with external RADIUS (accounting) servers to generate CDRs and provide billing services requires.

QoS Accounting Configuration

To configure the account configuration and account servers:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type account-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# account-config
ACMEPACKET(account-config)#
```

4. To configure account server parameters (a subset of the account configuration parameters, type account-servers and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(account-config)# account-servers
ACMEPACKET(account-server)#
```

The following example shows both the account config and account server parameters.

```
account-config
        hostname                        acctserver1
        port                            1813
        strategy                        Hunt
        state                           enabled
        max-msg-delay                   60
        max-wait-failover               100
        trans-at-close                  disabled
        generate-start                  OK
        generate-interim
                                        OK
                                        Reinvite-Response
account-server
                hostname                        192.168.2.2
                port                            1813
                state                           enabled
                min-round-trip                  100
                max-inactivity                  100
                restart-delay                   100
                bundle-vsa                      enabled
                secret                          testing
                NAS-ID                          acme-accounting
last-modified-date              2005-01-15 02:23:42
```

Account Configuration

You set the account configuration parameters to indicate where you want accounting messages sent, when accounting messages you want them sent, and the strategy you want used to select account servers.

To configure the account configuration:

1. **hostname**—Enter a name for the host associated with the Oracle Enterprise Session Border Controller in hostname (FQDN) format. The default value is the name of the local host.

The value you enter here must match the configured physical interface's operation type control or maintenance, to determine on which network to send RADIUS messages.

2. **port**—Enter the number of the UDP port associated with the Oracle Enterprise Session Border Controller from which RADIUS messages are sent.The default value is 1813. The valid range is:

   - Minimum—1025
   - Maximum—65535

3. **strategy**—Indicate the strategy you want used to select the accounting servers to which the Oracle Enterprise Session Border Controller will send its accounting messages. The default value is hunt. The following table lists the available strategies:

   - **hunt**—Selects accounting servers in the order in which they are listed.

     If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers
   - **failover**—Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
   - **roundrobin**—Selects each accounting server in order, distributing the selection of each accounting server evenly over time.
   - **fastestrtt**—Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
   - **fewestpending**—Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the Oracle Enterprise Session Border Controller).

4. **state**—Enable this parameter if you want the account configuration active on the system. Disable it if you do not want the account configuration active on the system. The default value is enabled. The valid values are:

   - enabled | disabled

5. **max-msg-delay**—Indicate the length of time in seconds that you want the Oracle Enterprise Session Border Controller to continue trying to send each accounting message. During this delay, the Oracle Enterprise Session Border Controller can hold a generic queue of 4096 messages. The default value is 60.

   - Minimum—zero (0)
   - Maximum—4294967295

6. **max-wait-failover**—Indicate the maximum number of accounting messages the Oracle Enterprise Session Border Controller can store its message waiting queue for a specific accounting server, before it is considered a failover situation.

   Once this value is exceeded, the Oracle Enterprise Session Border Controller attempts to send it accounting messages, including its pending messages, to the next accounting server in its configured list. The default value is 100. The valid range is:

   - Minimum—1
   - Maximum—4096

7. **trans-at-close**—Disable this parameter if you do not want to defer the transmission of message information to the close of a session. Enable it if you want to defer message transmission. The default value is disabled. The valid values are:

   - **disabled**—The Oracle Enterprise Session Border Controller transmits accounting information at the start of a session (Start), during the session (Interim), and at the close of a session (Stop). The transmitted accounting information for a single session might span a period of hours and be spread out among different storage files.
   - **enabled**—Limits the number of files on the Oracle Enterprise Session Border Controller used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.

8. **generate-start**—Select the type of SIP event that triggers the Oracle Enterprise Session Border Controller to transmit a RADIUS Start message. The default value is ok. The valid values are:
   - **start**—RADIUS Start message should not be generated
   - **invite—**RADIUS Start message should be generated once the Oracle Enterprise Session Border Controller receives a SIP session INVITE.
   - **ok**—RADIUS Start message is generated once the Oracle Enterprise Session Border Controller receives an OK message in response to an INVITE.
9. **generate-interim**—Retain the default value reinvite-response to cause the Oracle Enterprise Session Border Controller to transmit a RADIUS Interim message. (A RADIUS Interim message indicates to the accounting server that the SIP session parameters have changed.)

   To disable interim message generation, enter a pair of quotes as the value for this parameter. Otherwise, select one or more than one of the following values:
   - **ok**—RADIUS Interim message is generated when the Oracle Enterprise Session Border Controller receives an OK message in response to an INVITE.
   - **reinvite**—RADIUS Interim message is generated when the Oracle Enterprise Session Border Controller receives a SIP session reINVITE message.
   - **reinvite-response**—RADIUS Interim message is generated when the Oracle Enterprise Session Border Controller receives a SIP session reINVITE and responds to it (for example, session connection or failure).
   - **reinvite-cancel**—RADIUS Interim message is generated when the Oracle Enterprise Session Border Controller receives a SIP session reINVITE, and the Reinvite is cancelled before the Oracle Enterprise Session Border Controller responds to it.
10. **account-server**—Create the account server list to store accounting server information for the account configuration. Each account server can hold 100 accounting messages.

    Account server entries are specific to the account configuration. They cannot be viewed or accessed for editing outside of the account configuration.

    Note: RADIUS will not work if you do not enter one or more servers in a list.

Account Server

You must establish the list of servers to which the Oracle Enterprise Session Border Controller can send accounting messages.

1. hostname—Name of the host associated with the account server as an IP address.
2. port—Enter the number of the UDP port associated with the account server to which RADIUS messages are sent. The default value is 1813. The valid range is:
   - Minimum—1025
   - Maximum—65535
3. state—Enable or disable the account servers on the system. The default value is enabled. The valid values are:
   - enabled | disabled
4. min-round-trip—Indicate the minimum round trip time of an accounting message in milliseconds. The default value is 250. The valid range is:
   - Minimum—10
   - Maximum—5000

   A round trip consists of the following:

   The system sends an accounting message to the account server.

   The account server processes this message and responds back to the Oracle Enterprise Session Border Controller.

If the fastest RTT is the strategy for the account configuration, the value you enter here can be used to determine an order of preference (if all the configured account servers are responding in less than their minimum RTT).

5. max-inactivity—Indicate the length of time in seconds that you want the Oracle Enterprise Session Border Controller with pending accounting messages to wait when it has not received a valid response from the target account server. The default value is 60. The valid range is:

   - Minimum—1
   - Maximum—300

   Once this timer value is exceeded, the Oracle Enterprise Session Border Controller marks the unresponsive account server as disabled in its failover scheme. When a server connection is marked as inactive, the Oracle Enterprise Session Border Controller attempts to restart the connection and transfers pending messages to another queue for transmission. RADIUS messages might be moved between different account servers as servers become inactive or disabled.

6. restart-delay—Indicate the length of time in seconds you want the Oracle Enterprise Session Border Controller to wait before resending messages to a disabled account server. The default value is 30. The valid range is:

   - Minimum—1
   - Maximum—300

7. bundle-vsa—Retain the default enabled if you want the account server to bundle the VSAs within RADIUS accounting messages. Enter disabled if you do not want the VSAs to be bundled. (Bundling means including multiple VSAs within the vendor value portion of the message.) The valid values are:

   - enabled | disabled

   In a bundled accounting message, the RADIUS message type is vendor-specific, the length is determined for each individual message, and the vendor portion begins with a 4-byte identifier, and includes multiple vendor type, vendor length, and vendor value attributes.

8. secret—Enter the secret passed from the account server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IPv4 address of the received packet.

9. NAS-ID—Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the Oracle Enterprise Session Border Controller for the transmittal of accounting messages.

   The remote server to which the account configuration sends messages uses at least one of two potential pieces of information for purposes of identification. The Oracle Enterprise Session Border Controller accounting messages always includes in the first of these:

   - Network Access Server (NAS) IP address (the IP address of the Oracle Enterprise Session Border Controller's SIP proxy)
   - NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

Whitelists for SIP

Oracle Enterprise Session Border Controller by default ignores and passes-through unknown SIP headers and URI parameters. However, some operators require that the Oracle Enterprise Session Border Controller only accept messages with headers and URI parameters complying with those supported by their internal equipment. This section describes the use of whitelists to control unknown headers and parameters in request and response traffic.

What is a Whitelist

A whitelist is an approved list of entities for which equipment provides particular privileges, access, and recognition. The Oracle Enterprise Session Border Controller can use configured whitelist profiles to control and accept specific inbound SIP headers and URI parameters that are being passed-through the Oracle Enterprise Session Border

Controller. When you configure this feature, the Oracle Enterprise Session Border Controller rejects requests not matching the configured profile, or removes the unspecified headers or URI parameters not in the configured profile.

Whitelists Configuration

You can configure whitelist profiles or rules that allow the Oracle Enterprise Session Border Controller to only accept inbound SIP headers and URI parameters that are configured in this whitelist, using the parameter allowed-elements-profile. You can configure the settings for this parameter using the ACLI interface at session-router>enforcement-profile. Since the enforcement-profile object also pertains to session agents, realms, and SIP interfaces, you can also apply the profiles you configure to these remote entities using the ACLI interface at session-router>session-agent, session-router>sip-interface, and media-manager>realm-config.

In the following configuration example, it is assumed that your baseline configuration passes SIP traffic, with the Oracle Enterprise Session Border Controller in the role of an Access SBC. Use this procedure to configure a whitelist for the session router and optionally apply the specific whitelists to the session agent and SIP interface, as well as the media manager's realm configuration.

To configure a whitelist for the session router:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type allowed-elements-profile and press Enter to access the enforcement profile-related parameters.

```
ACMEPACKET(session-router)# allowed-elements-profile
ACMEPACKET(allowed-elements-profile)#
```

4. Type allowed-elements-profile and press Enter to access the whitelist-related parameters. The system prompt changes to let you know that you can begin configuring individual parameters for this object.

```
ACMEPACKET(enforcement-profile)# allowed-elements-profile
ACMEPACKET(allowed-elements-profile)#
```

5. name — Enter a unique name for the whitelist you are creating. This name can be referenced when configuring the enforcement-profiles for session-agent, SIP interface, and realm-config.

```
ACMEPACKET(allowed-elements-profile)# name whitelist1
```

6. description — Enter a description that explains the purpose of creating this whitelist. You can use any alpha-numeric characters when entering the description.

```
ACMEPACKET(allowed-elements-profile)# description Basic Whitelist
```

7. Type rule-sets and press Enter to specify the rules to match against specific incoming SIP headers and/or URI parameters. The system prompt changes to let you know that you can begin configuring individual parameters for this object.

```
ACMEPACKET(allowed-elements-profile)# rule-sets
ACMEPACKET(rule-sets)#
```

8. unmatched-action — Select the action for the Oracle Enterprise Session Border Controller to perform when a header does not exist in an incoming message. The default value is reject. The valid values are:

   - reject — Rejects all incoming messages that do not contain a header.
   - delete — Deletes all incoming message that do not contain a header.

   Note: This parameter applies to non-matching header names only (not non-matching URI parameters).

```
ACMEPACKET(rule-sets)# unmatched action delete
```

9. msg-type — Specify the type of messages for which the Oracle Enterprise Session Border Controller applies this whitelist configuration. The default value is any. The valid values are:

   - any — Applies to all incoming messages.
   - request — Applies to only incoming REQUEST messages.

- response — Applies to only incoming RESPONSE messages.

```
ACMEPACKET(rule-sets)# msg-type any
```

10. methods — Enter the packet method(s), separated by a comma, for which this whitelist is enforced. Packet methods include, INVITE, OPTIONS, ACK, BYE, etc. If this field is left blank, the whitelist applies to all packet methods. You can enter up to a maximum of 255 characters.

```
ACMEPACKET(rule-sets)# methods INVITE,ACK,BYE
```

11. logging — Select whether or not an incoming message is written to a log file called matched.log when the message contains an element not specified in the whitelist. The default value is disabled. The valid values are:

- enabled | disabled

```
ACMEPACKET(rule-sets)# logging enabled
```

The matched.log contains information about the timestamp, received/sent Oracle Enterprise Session Border Controller network-interface, IP address/port from which it was received or being sent from, and which peer IP address/port it was received from or sent to. The log also specifies the request URI (if applicable), and the From, To, and Contact headers in the message, as well as which rule triggered the log action. An example of the log output of the matched.log file is as follows:

```
Dec 17 14:17:54.526 On [0:0]192.168.1.84:5060 sent to 192.168.1.60:5060
allowed-elements-profile[whitelist1(reject)]
INVITE sip:service@192.168.1.84:5060 SIP/2.0
From: sipp <sip:+2125551212@192.168.1.60:5060>;tag=3035SIPpTag001
To: sut <sip:service@192.168.1.84>
Contact: sip:sipp@192.168.1.60:5060
```

The header-rule object consists of 6 parameters that make up the header-rule:

- header-name
- unmatched-action
- allow-header-param
- allow-uri-param
- allow-uri-user-param
- allow-uri-header-name

  You can configure an unlimited number of header-rules on the Oracle Enterprise Session Border Controller that you can apply to your network. Use the following parameters to configure a header-rule that the Oracle Enterprise Session Border Controller uses to control which incoming messages it allows.

12. header-rule — This object allows you to configure, as part of the whitelist, multiple parameters which make up the header rule that the Oracle Enterprise Session Border Controller allows from incoming messages. Header-rules do NOT have to be in any specific order. The system prompt changes to let you know that you can begin configuring individual parameters for this object.

```
ACMEPACKET(rule-set)# header-rule
ACMEPACKET(allowed-header-rule)#
```

13. header-name — Enter the name of the header in the whitelist that the Oracle Enterprise Session Border Controller allows from incoming messages. It is case-insensitive and supports abbreviated forms of header names. For example, “Via”, “via”, or “v” all match against the same header. A header name of “request-uri” refers to the request URI of requests, while a header name of * applies to any header-type not matched by any other header-rule. The default value is *. This default value provides the ability to have header-rules for commonly known headers that remove unknown parameters, but leave unknown headers alone.

```
ACMEPACKET(allowed-header-rule)# header-name Contact
```

14. unmatched-action — Select the action for the Oracle Enterprise Session Border Controller to perform when an incoming header’s parameters do not match the relevant allowed parameters specified for this header-name. The default value is reject. The valid values are:

- reject — Rejects all incoming messages that have header parameters that do not match the parameters specified in this header-name.

- delete — Deletes all incoming messages that have header parameters that do not match the parameters specified in this header-name.

  Note: This parameter applies to non-matching header names only (not non-matching URI parameters).

```
ACMEPACKET(allowed-header-rule)# unmatched-action delete
```

15. allow-header-param — Enter the header parameter that the Oracle Enterprise Session Border Controller allows from the headers in incoming messages. You can enter up to 255 characters, including a comma (,), semi-colon (;), equal sign (=), question mark (?), at-symbol (@), backslash (\), or plus sign (+). The default value is *, which allows all header parameters to pass through. If you leave this field empty, no header parameters are allowed.

```
ACMEPACKET(allowed-header-rule)# allow-header-param *
```

16. allow-uri-param — Enter the URI parameter that the Oracle Enterprise Session Border Controller allows from the headers in incoming messages. You can enter up to 255 characters, including a comma (,), semi-colon (;), equal sign (=), question mark (?), at-symbol (@), backslash (\), or plus sign (+). The default value is *, which allows all URI parameters to pass through. If you leave this field empty, no URI parameters are allowed.

```
ACMEPACKET(allowed-header-rule)# allow-uri-param *
```

17. allow-uri-user-param — Enter the URI user parameter that the Oracle Enterprise Session Border Controller allows from the headers in incoming messages. You can enter up to 255 characters, including a comma (,), semi-colon (;), equal sign (=), question mark (?), at-symbol (@), backslash (\), or plus sign (+). The default value is *, which allows all URI user parameters to pass through. If you leave this field empty, no URI user parameters are allowed.

```
ACMEPACKET(allowed-header-rule)# allow-uri-user-param *
```

18. allow-uri-header-name — Enter the URI header name that the Oracle Enterprise Session Border Controller allows from the headers in incoming messages. You can enter up to 255 characters, including a comma (,), semi-colon (;), equal sign (=), question mark (?), at-symbol (@), backslash (\), or plus sign (+). The default value is *, which allows all URI header name parameters to pass through. If you leave this field empty, no URI header name parameters are allowed.

```
ACMEPACKET(allowed-header-rule)# allow-uri-header-name *
```

19. Save your work using the ACLI done command.

Configuration Exception

There are specific instances in incoming Request-URI messages where the Oracle Enterprise Session Border Controller ignores specific parameters and automatically adds header-rules.

In a Request-URI, all parameters are URI parameters, and URI headers are not allowed. If you define values for the “allow-header-param”, “allow-uri-header-name”, and “allow-uri-param”, the Oracle Enterprise Session Border Controller ignores these parameters in the Request-URI. Instead the Oracle Enterprise Session Border Controller automatically adds header-rules for incoming “Via”, “From”, To”, “Call-ID”, and “CSeq” messages. These are explicit header rules and cannot be deleted. Each header-rule in a Request-URI has parameters populated with the value of *. If required, a user can change the header-rule parameter values with the values identified in the following table.

Header Rule	Applicable Parameter	Required Value(s)
Via	allow-header-param	• branch • received • rport
From	allow-header-param	• tag
To	allow-header-param	• tag
Call-ID	allow-header-param	No restrictions
CSeq	allow-header-param	No restrictions

Verify Whitelist Configuration

After you have configured and saved a whitelist on the Oracle Enterprise Session Border Controller, you can use the existing verify-config command at the top level prompt to verify the saved configuration: For example:

```
ACMEPACKET# verify-config
```

This command checks for errors in the Oracle Enterprise Session Border Controller configuration. Whitelist configuration errors, specifically related to the enforcement-profile object, also display in the output of this command if applicable. The whitelist configuration error(s) display if any references to the allowed-element-profiles are improperly configured. If an error(s) exist, the following message displays:

```
-------------------------------------------------------------------------
ERROR: enforcement-profile [ep] contains a reference to an allowed-
enforcement-profile [abc] that does not exist
-------------------------------------------------------------------------
```

How Whitelists Work

Whitelists allow you to customize which SIP signaling messages to allow into your network and which messages to reject or delete. In the flow of SIP traffic to/from the Oracle Enterprise Session Border Controller, the Oracle Enterprise Session Border Controller matches any received request or response, in or out of a dialog against the configured allowed list, and rejects or deletes the non-matching element based on the actions specified in the whitelist configuration.

For responses, the Oracle Enterprise Session Border Controller does not reject the message if a header or parameter is not found in the allowed list, even if the action is set to reject. Instead it deletes the offending parameter or header. In addition, if the message is a request of the method type ACK, PRACK, CANCEL or BYE, it deletes all unmatched elements, but does not reject the request, even if the action was configured to reject.

The whitelist verification performs for any method; however you can narrow this list to operate only on specific methods by defining them in the methods parameter of the configuration.

Whitelist verification occurs when a request or response is received by the Oracle Enterprise Session Border Controller, but only after the Oracle Enterprise Session Border Controller has processed the inbound header manipulation rule (HMR), network management controls (NMC), Resource-Priority header (RPH), and monthly-minutes checking.

The Oracle Enterprise Session Border Controller responds to requests which have non-matching headers or parameters configured with an action of reject, with a "403 Forbidden" response by default. You can use a local-response event, allowed-elements-profile-rejection, to override the default reject status code and reason phrase.

The configured whitelist operates transparently on headers that have multiple URIs or multiple header values within a single header (header values separated by a comma).

Parameter parsing operates only on parameters that it can identify. For parameters that can not be parsed, for example an invalid URI (e.g. <sip:user@host.com&hp=val>), the Oracle Enterprise Session Border Controller ignores this URI header parameter value of "hp" since it is not contained within a valid URI. Even though it would appear to be a URI header parameter, URI headers must come after URI parameters. Parameter matching does not occur if the headers and parameters in the URI are not well-formed. The Oracle Enterprise Session Border Controller does not remove the parameter since it cannot identify it.

Whitelist Learning

You can build your whitelist configuration based on the learning capabilities of the Oracle Enterprise Session Border Controller. When you enable the Oracle Enterprise Session Border Controller learning mode, it acquires the knowledge of the allowable elements (headers and parameters) currently incoming to your network. The Oracle Enterprise Session Border Controller collects the information about the headers received and the parameters that exist within each header. The information continues to be gathered until you disable the learning mode.

Once you disable the learning mode, the Oracle Enterprise Session Border Controller prompts you to enter a name for the allowed-elements-profile. If the profile name you entered does not exist, the captured information is written to the new allowed-elements-profile configuration. The administrator can then make changes to the configuration as applicable, save the configuration, and apply it to a logical remote entity.

The new allowed-elements-profile does not contain any wildcard rules. The Oracle Enterprise Session Border Controller cannot generate wildcard headers and parameters during the learning mode. The Methods object is populated from the list of methods seen by the Oracle Enterprise Session Border Controller while learning.

Note: Oracle recommends running the learning mode during off-peak and/or light traffic times. This mode can operate in conjunction with the execution of an allowed-elements-profile. The learning occurs just before any configured allowed-elements-profile configuration.

Whitelist Learning Configuration

The ACLI interface provides two commands that allow a Superuser to start and stop whitelist learning on the Oracle Enterprise Session Border Controller:

Command	Description
start <argument> <options>	Starts whitelist learning on the Oracle Enterprise Session Border Controller. You must specify the argument learn-allowed-elements with this command to start the learning operation. Optionally, you can use method, msg-type, and params after the argument.
stop <argument> <identifier>	Stops the whitelist learning on the Oracle Enterprise Session Border Controller and writes the learned configuration to the editing configuration on the Oracle Enterprise Session Border Controller where it is saved and activated. You must specify the argument learn-allowed-elements with this command to stop the learning operation. You must specify a unique identifier that identifies the allowed-elements-profile name. If you specify an identifier name that already exists as a profile, the ACLI returns an error message and prompts you to enter a different name.

You can use these commands at the top level ACLI prompt as required on the Oracle Enterprise Session Border Controller.

You use these commands with the argument, learn-allowed-elements to start/stop the whitelist learning feature. By default, the learning mode creates a single rule-set under which all of the headers and their respective parameters are stored.

For example:

```
ACMEPACKET# start learn-allowed-elements
Learning mode for allowed-elements-profile started.
```

In the above example, “start” is the top level ACLI command and learn-allowed-elements is the operation being performed.

Optionally, you can specify [method], [msg-type], and [params] in any order, for the Oracle Enterprise Session Border Controller to learn specific rule-set elements from incoming messages and save them to the whitelist configuration.

For example:

```
ACMEPACKET# start learn-allowed-elements method msg-type params
```

The method option creates a new rule-set per unique method. The "msg-type" option creates a new rule-set per unique message-type seen. The "params" option performs URI and header parsing to examine parameters within the message. By default, parameter parsing is disabled.

To start the whitelist learning feature:

In Superuser mode, at the top level ACLI prompt, type start learn-allowed-elements and press Enter.

```
ACMEPACKET# start learn-allowed-elements
```

The following message displays:

```
Learning mode for allowed-elements-profile started.
```

To specify the elements of rule-sets for whitelists:

In Superuser mode, at the top level ACLI prompt, type start learn-allowed-elements method msg-type params and press Enter.

```
ACMEPACKET# start learn-allowed-elements method msg-type params
```

The following message displays:

```
Learning mode for allowed-elements-profile started.
```

Note: If you try to start a whitelist learning operation while another learning operation is already running, the following message displays:

```
Learning mode restarted without saving
Learning mode for allowed-elements-profile started.
```

To stop the whitelist learning feature:

In Superuser mode, at the top level ACLI prompt, type stop learn-allowed-elements <identifier>, where <identifier> is the allowed-elements-profile name, and press Enter.

```
ACMEPACKET# stop learn-allowed-elements whitelist1
```

The following message displays:

```
Learning mode for allowed-elements-profile stopped.
```

If you specify an identifier name that already exists as a profile, the ACLI returns an error message and prompts you to enter a different name.

Rejected Messages Monitoring

Whitelists, when configured on the Oracle Enterprise Session Border Controller, control whether or not the Oracle Enterprise Session Border Controller allows unknown headers and URI parameters to be accepted in incoming request and response traffic. When the Oracle Enterprise Session Border Controller rejects messages according to the whitelist, the rejected messages are logged to a file called “matched.log” if logging is set to enabled. You can open and view the log when required to view the rejected messages.

In addition to the rejected messages being logged to the “matched.log” file, the rejected messages are also sent through a burst counter that keeps track of the amount of messages rejected. You can enter the show sipd to display the number of rejected messages. The counter is titled Rejected Message.

17

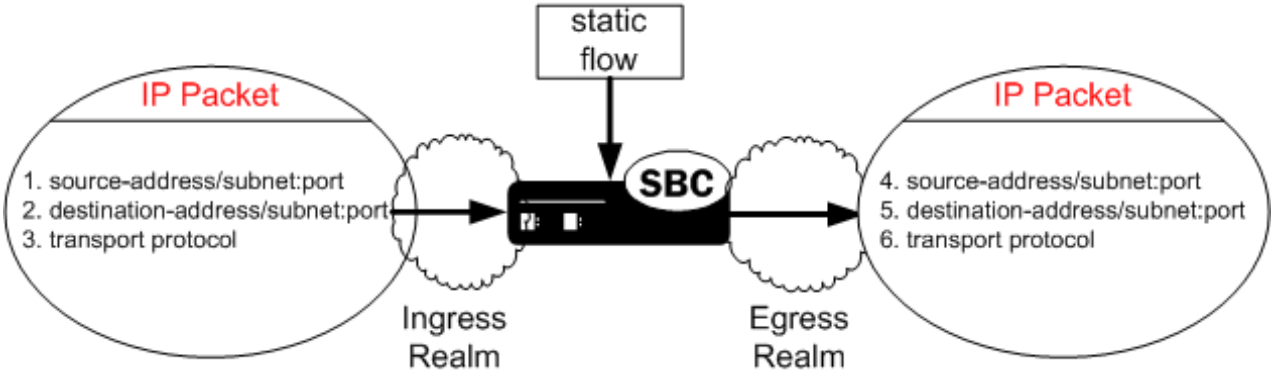
Static Flows

This chapter describes the Oracle Enterprise Session Border Controller ’s static flows feature. Static flows allow network traffic that matches specific criteria to pass through the Oracle Enterprise Session Border Controller unrestricted. This feature lets you steer traffic toward a particular destination based on its original characteristics. Static flows can range from being widely accessible to very restrictive, depending on the values you establish. Static flows are used for transporting a variety of signaling messages through the Oracle Enterprise Session Border Controller to achieve vendor interoperability. The Oracle Enterprise Session Border Controller supports the following types of Static flows:

- IPv6 to IPv6 flows
- IPv4 to IPv6 flows
- IPv4 to IPv4 flows

About Static Flows

The static flow element explicitly writes entries into the IP routing table. These entries are persistent and are not deleted as calls are set up and broken down. Refer to the following diagram to understand how a static flow works.



A static flow entry watches for traffic with specific criteria on a specified ingress realm; that traffic consists of the following criteria:

- The packet enters the Oracle Enterprise Session Border Controller on the specified ingress realm.
- The packet contains matching source address, subnet, and port criteria, field 1.
- The packet contains matching destination address, subnet, and port criteria, field 2.
- The packet contains a matching transport protocol, field 3.

If the above conditions are met, then the Oracle Enterprise Session Border Controller does the following:

- The IPv4 traffic is forwarded out of the Oracle Enterprise Session Border Controller on the specified egress realm.
- The configured source address, subnet, and port criteria are written to the exiting packet, field 4.
- The configured destination address, subnet, and port criteria are written to the exiting packet, field 5.
- The original transport protocol and its contents remain unchanged as the packet exits into the egress realm.

IPv6 / IPv4 Translations

The ingress or egress traffic type, whether IPv4 or IPv6, must match the configuration of the realm where attached, as **in-realm-id** or **out-realm-id**. A realm and IP version configuration mismatch results in an error message and log entry at error level.

IPv6 to IPv4 flows exit the Oracle Enterprise Session Border Controller with the prefix ::fff:0:0:0/96. They may be written as ::ffff:0:a.b.c.d, where a.b.c.d refers to an IPv6-enabled node.

While IPv4 addresses can be translated into IPv6 addresses, IPv6 address can not be translated to IPv4.

About Network Address Translation ALG

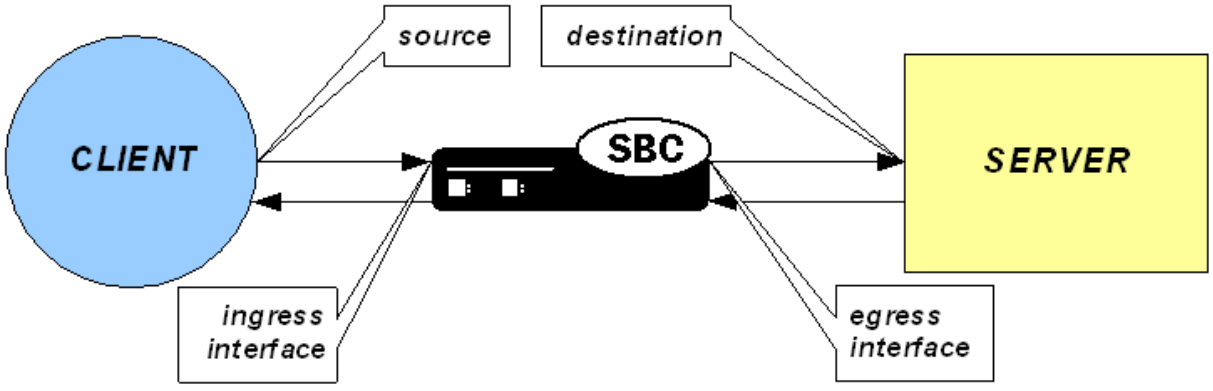
The Oracle Enterprise Session Border Controller supports Network Address and Port Translation (NAPT) and Trivial File Transfer Protocol (TFTP) functionality over media interfaces, collectively known as Network Address Translation (NAT) ALG. The NAT ALG feature is implemented as an extension of the static flow feature.

In some applications, the Oracle Enterprise Session Border Controller acts as an intermediary device, positioned between endpoints located in an access network and application servers located in a backbone network. The Oracle Enterprise Session Border Controller 's NAT ALG feature enables these endpoints to use non-VoIP protocols, such as TFTP and HTTP, to access servers in a provider's backbone network to obtain configuration information.

NAT ALG parameters support RTC and can be dynamically reconfigured. The active NAT ALG configuration can be replicated on the standby SD in an HA configuration.

NAPT

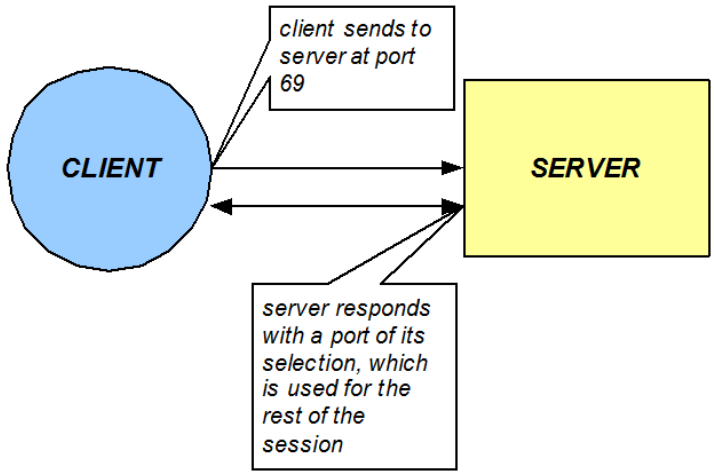
The NAPT ALG functionality is the same as that found in commercially available enterprise and residential NAT devices. The Oracle Enterprise Session Border Controller watches for packets entering a media interface that match source and destination IP address criteria. Matching packets are then redirected out of the egress interface, through a specified port range, toward a destination address.



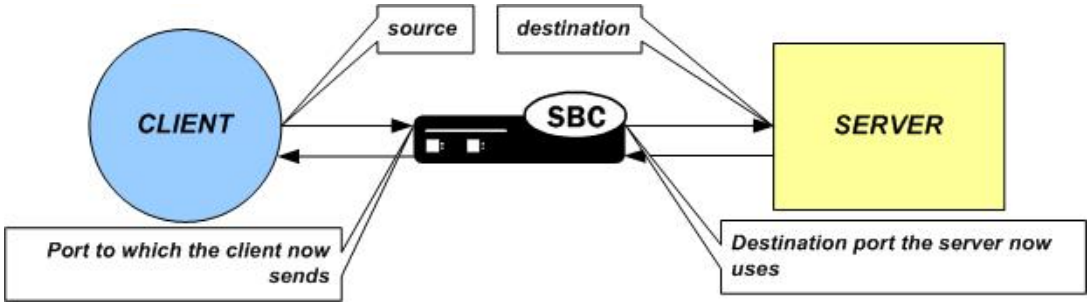
TFTP

The TFTP ALG is implemented as an extension of the NAT ALG. It works slightly differently than traditional NAPT. In a TFTP session, the first packet is sent from a source endpoint to port 69 on the TFTP server. The TFTP server

responds from another port. This port, from which the TFTP response originates, is used for the remainder of the TFTP session.



To act as a TFTP ALG, the Oracle Enterprise Session Border Controller will latch on the first return packet from the server to learn the server's port. The ingress-side destination port of the Oracle Enterprise Session Border Controller is changed to reflect the new communications port for the TFTP session. This process takes place without any user intervention.



Configuring Static Flows

This section explains how to configure static flows. It also provides sample configurations for your reference. You can configure static flows with or without NAT ALG. If you configure static flows with NAT ALG, you can choose NAPT or TFTP as the ALG type.

Basic Static Flow Configuration Overview

This section outlines the basic static flow configuration, without NAT ALG. You configure static flows by specifying ingress traffic criteria followed by egress re-sourcing criteria.

When configuring static flows, the following conventions are used:

- An address of 0.0.0.0 matches all addresses. This token is used as the wildcard for both IPv4 and IPv6 static flows
- Enclose the address portion of an IPv6 address in brackets: [7777::11]/64:5000
- Not specifying a port implies all ports.
- Not specifying a subnet mask implies a /32, matching for all 32 bits of the IPv4 address , or a /128 matching for all 128 bits of the IPv6 address.

1. Set the static flows' incoming traffic-matching criteria. First set the ingress realm where you expect to receive traffic that will be routed via a static flow. Second, set the traffic's source IP address, source subnet, and source port or port range criteria. Third, set the traffic's destination IP address, destination subnet, and destination port criteria. This is usually an external address on theOracle Enterprise Session Border Controller.

2. Set the criteria that describes how traffic should be translated on the egress side of the Oracle Enterprise Session Border Controller. First set the egress realm where you want to send the traffic to be routed by this static flow. Second, set the traffic's source IP address, source subnet, and source port or port range criteria. This is usually an external address on the Oracle Enterprise Session Border Controller. Third, set the traffic's destination IP address, destination subnet, and destination port criteria.
3. Set the protocol this static flow entry acts upon. This type of packet, as the payload of the IP packet, remains untouched as traffic leaves the Oracle Enterprise Session Border Controller . Specifying a layer 4 protocol here acts as another criteria to filter against for this static flow.

   The combination of entries in the ingress realm, ingress source address, ingress destination address, and protocol fields must be unique. For bidirectional traffic, you need to define a separate static flow in the opposite direction.

Static Flow Configuration

This section describes how to configure the static-flow element using the ACLI.

The ingress IP address criteria is set first. These parameters are applicable to traffic entering the ingress side of the Oracle Enterprise Session Border Controller .

- in-realm-id—The access realm, where endpoints are located.
- in-source—The source network in the access realm where the endpoints exist. This parameter is entered as an IP address and netmask in slash notation to indicate a range of possible IP addresses.
- in-destination—The IP address and port pair where the endpoints send their traffic. This is usually the IP address and port on a Oracle Enterprise Session Border Controller physical interface that faces the access realm.

The egress IP address criteria is entered next. These parameters determine how traffic is re-sourced as it leaves the Oracle Enterprise Session Border Controller and enters the backbone network.

- out-realm-id—The backbone realm, where servers are located.
- out-source—The IP address on the physical interface of the Oracle Enterprise Session Border Controller where traffic exits the Oracle Enterprise Session Border Controller into the backbone realm. Do not enter a port for this parameter.
- out-destination—The IP address and port pair destination of the traffic. This is usually a server in the backbone realm.
- protocol—The protocol associated with the static flow. The protocol you choose must match the protocol in the IPv4 header. Valid entries are TCP, UDP, ICMP, ALL.

The type of NAT ALG, if any.

- alg-type—The type of NAT ALG. Set this to NAPT, TFTP, or none.

The port range for port re-sourcing as traffic affected by the NAT ALG exits the egress side of the Oracle Enterprise Session Border Controller is set next. (Not applicable if alg-type is set to none.)

- start-port—The starting port the NAT ALG uses as it re-sources traffic on the egress side of the Oracle Enterprise Session Border Controller .
- end-port—The ending port the NAT ALG uses as it re-sources traffic on the egress side of the Oracle Enterprise Session Border Controller .

The flow timers are set next. (Not applicable if alg-type is set to none.)

- flow-time-limit—Total session time limit in seconds. The default is 0; no limit.

  Note: Note that the static flow-time-limit must have a value larger than initial-guard-timer and subsq-guard-timer for static flows.
- initial-guard-timer—Initial flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit.
- susbsq-guard-timer—Subsequent flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit.

Finally, you can set the optional bandwidth policing parameter for static flows (with or without NAT ALG applied).

- average-rate-limit—Sustained rate limit in bytes per second for the static flow and any dynamic ALG flows. The default is 0; no limit.

To configure static flow:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type static-flow and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# static-flow
```

From this point, you can configure media policing parameters.

4. in-realm-id—Enter the ingress realm or interface source of packets to match for static flow translation. This in-realm-id field value must correspond to a valid identifier field entry in a realm-config. This is a required field. Entries in this field must follow the Name Format.
5. in-source—Enter the incoming source IP address and port of packets to match for static flow translation. IP address of 0.0.0.0 matches any source address. Port 0 matches packets received on any port. The port value has no impact on system operation if either ICMP or ALL is the selected protocol. This parameter takes the format:

   in-source <ip-address>[:<port>]

   The default value is 0.0.0.0. The valid port range is:

   - Minimum—0
   - Maximum—65535
6. in-destination—Enter the incoming destination IP address and port of packets to match for static-flow translation. An IP address of 0.0.0.0 matches any source address. Port 0 matches packets received on any port. The port value has no impact on system operation if either ICMP or ALL is the selected protocol. The in-source parameter takes the format:

   in-destination <ip-address>[:<port>]

   The default value is 0.0.0.0. The valid port range is:

   - Minimum—0
   - Maximum—65535
7. out-realm-id—Enter the defined realm where traffic leaving this NAT ALG exits theOracle Enterprise Session Border Controller .
8. out-source—Enter the egress IPv4 address. This is the IPv4 address of the network interface where traffic subject to the NAT ALG you are defining leaves the Oracle Enterprise Session Border Controller . Do not enter a port number for this parameter. The default value is 0.0.0.0.
9. out-destination—Enter the IPv4 address and port number of the server or other destination to which traffic is directed. The default value is 0.0.0.0. The valid port range is:

   - Minimum—0
   - Maximum—65535
10. protocol—Enter the protocol this NAPT ALG acts upon. The default value is UDP. The valid values are:

    - TCP | UDP | ICMP | ALL
11. alg-type—Enter the type of NAT ALG to use. The default value is none. The valid values are:

    - none—No dynamic ALG functionality
    - NAPT—Configure as NAPT ALG
    - TFTP—Configure as TFTP ALG
12. start-port—Enter the beginning port number of the port range that the Oracle Enterprise Session Border Controller allocates on the egress side for flows that this NAPT ALG redirects. The default value is 0. The valid range is:

- Minimum—0, 1025
- Maximum—65535

13. end-port—Enter the ending port number of the port range that the Oracle Enterprise Session Border Controller allocates on the egress side for flows that this NAPT ALG redirects. The default value is 0. The valid range is:

    - Minimum—0, 1025
    - Maximum—65535

14. flow-time-limit—Enter the total time limit for a flow in seconds. A value of 0 means there is no limit. The valid range is:

    - Minimum—0
    - Maximum—999999999

15. initial-guard-timer—Enter the initial guard timer value in seconds. A value of 0 means there is no limit. The valid range is:

    - Minimum—0
    - Maximum—999999999

16. subsq-guard-timer—Enter the subsequent guard timer value in seconds. A value of 0 means there is no limit. The valid range is:

    - Minimum—0
    - Maximum—999999999

17. average-rate-limit—Enter a maximum sustained rate limit in bytes per second. The default value is 0; no limit. The valid range is:

    - Minimum—0
    - Maximum—125000000

    The following example shows a static-flow configuration element configured for a NAPT ALG.

```
        in-realm-id                    access
        in-source                      172.16.0.0/16
        in-destination                 172.16.1.16:23
        out-realm-id                   backbone
        out-source                     192.168.24.16
        out-destination                192.168.24.95:23
        protocol                       TCP
        alg-type                       NAPT
        start-port                     11000
        end-port                       11999
        flow-time-limit                0
        initial-guard-timer            60
        subsq-guard-timer              60
        average-rate-limit             0
```

18

High Availability Nodes

Oracle Enterprise Session Border Controller s can be deployed in pairs to deliver high availability (HA). Two Oracle Enterprise Session Border Controller s operating in this way are called an HA node. Over the HA node, media and call state are shared, keeping sessions/calls from being dropped in the event of a failure.

Two Oracle Enterprise Session Border Controller s work together in an HA node, one in active mode and one in standby mode.

- The active Oracle Enterprise Session Border Controller checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it will hand over its role as the active system to the standby Oracle Enterprise Session Border Controller in the node.
- The standby Oracle Enterprise Session Border Controller is the backup system, fully synchronized with active Oracle Enterprise Session Border Controller s session status. The standby Oracle Enterprise Session Border Controller monitors the status of the active system so that, if needed, it can assume the active role without the active system having to instruct it to do so. If the standby system takes over the active role, it notifies network management using an SNMP trap.

In addition to providing instructions for how to configure HA nodes and their features, this chapter explains how to configure special parameters to support HA for all protocols.

Overview

To produce seamless switchovers from one Oracle Enterprise Session Border Controller to the other, the HA node uses shared virtual MAC and virtual IP addresses for the media interfaces in a way that is similar to VRRP (virtual router redundancy protocol). When there is a switchover, the standby Oracle Enterprise Session Border Controller sends out a gratuitous ARP messages using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted.

Within the HA node, the Oracle Enterprise Session Border Controller s advertise their current state and health to one another in checkpointing messages; each system is apprised of the other's status. Using Oracles HA protocol, the Oracle Enterprise Session Border Controller s communicate with UDP messages sent out and received on the rear interfaces.

The standby Oracle Enterprise Session Border Controller shares virtual MAC and IPv4 addresses for the media interfaces (similar to VRRP) with the active Oracle Enterprise Session Border Controller . Sharing addresses eliminates the possibility that the MAC and IPv4 address set on one Oracle Enterprise Session Border Controller in an HA node will be a single point of failure. The standby Oracle Enterprise Session Border Controller sends ARP requests using a utility IPv4 address and its hard-coded MAC addresses to obtain Layer 2 bindings.

The standby Oracle Enterprise Session Border Controller assumes the active role when:

- It has not received a checkpoint message from the active Oracle Enterprise Session Border Controller for a certain period of time.
- It determines that the active Oracle Enterprise Session Border Controller ’s health score has decreased to an unacceptable level.
- The active Oracle Enterprise Session Border Controller relinquishes the active role.

Establishing Active and Standby Roles

Oracle Enterprise Session Border Controller s establish active and standby roles in the following ways.

- If a Oracle Enterprise Session Border Controller boots up and is alone in the network, it is automatically the active system. If you then pair a second Oracle Enterprise Session Border Controller with the first to form an HA node, then the second system to boot up will establish itself as the standby automatically.
- If both Oracle Enterprise Session Border Controller s in the HA node boot up at the same time, they negotiate with each other for the active role. If both systems have perfect health, then the Oracle Enterprise Session Border Controller with the lowest HA rear interface IPv4 address will become the active Oracle Enterprise Session Border Controller . The Oracle Enterprise Session Border Controller with the higher HA rear interface IPv4 address will become the standby Oracle Enterprise Session Border Controller .
- If the rear physical link between the two Oracle Enterprise Session Border Controller s fails during boot up or operation, both will attempt to become the active Oracle Enterprise Session Border Controller . In this case, processing will not work properly.

Health Score

HA Nodes use health scores to determine their active and standby status. Health scores are based on a 100-point system. When a Oracle Enterprise Session Border Controller is functioning properly, its health score is 100.

Generally, the Oracle Enterprise Session Border Controller with the higher health score is active, and the Oracle Enterprise Session Border Controller with the lower health score is standby. However, the fact that you can configure health score thresholds builds some flexibility into using health scores to determine active and standby roles. This could mean, for example, that the active Oracle Enterprise Session Border Controller might have a health score lower than that of the standby Oracle Enterprise Session Border Controller , but a switchover will not take place because the active Oracle Enterprise Session Border Controller ’s health score is still above the threshold you configured.

Alarms are key in determining health score. Some alarms have specific health score value that are subtracted from the Oracle Enterprise Session Border Controller ’s health score when they occur. When alarms are cleared, the value is added back to the Oracle Enterprise Session Border Controller ’s health score.

You can look at a Oracle Enterprise Session Border Controller ’s health score using the ACLI show health command.

Switchovers

A switchover occurs when the active Oracle Enterprise Session Border Controller stops being the active system, and the standby system takes over that function. There are two kinds switchovers: automatic and manual.

Automatic Switchovers

Automatic switchovers are triggered without immediate intervention on your part. Oracle Enterprise Session Border Controller s switch over automatically in the following circumstances:

- When the active Oracle Enterprise Session Border Controller ’s health score of drops below the threshold you configure.
- When a time-out occurs, meaning that the active Oracle Enterprise Session Border Controller has not has not sent checkpointing messages to the standby Oracle Enterprise Session Border Controller within the allotted time.

  The active Oracle Enterprise Session Border Controller might not send checkpointing messages for various reasons such as link failure, communication loss, or advertisement loss. Even if the active Oracle Enterprise Session Border Controller has a perfect health score, it will give up the active role if it does not send a checkpoint

message or otherwise advertise its status within the time-out window. Then the standby Oracle Enterprise Session Border Controller takes over as the active system.

When an automatic switchover happens, the Oracle Enterprise Session Border Controller that has just become active sends an ARP message to the switch. This message informs the switch to send future messages to its MAC address. The Oracle Enterprise Session Border Controller that has just become standby ignores any messages sent to it.

Manual Switchovers

You can trigger a manual switchover in the HA node by using the ACLI notify berpd force command. This command forces the two Oracle Enterprise Session Border Controller s in the HA node to trade roles. The active system becomes standby, and the standby becomes active.

In order to perform a successful manual switchover, the following conditions must be met.

- The Oracle Enterprise Session Border Controller from which you trigger the switchover must be in one of the following states: active, standby, or becoming standby.
- A manual switchover to the active state is only allowed on a Oracle Enterprise Session Border Controller in the standby or becoming standby state if it has achieved full media, signaling, and configuration synchronization.
- A manual switchover to the active state is only allowed on a Oracle Enterprise Session Border Controller in the standby or becoming standby state if it has a health score above the value you configure for the threshold.

State Transitions

Oracle Enterprise Session Border Controller s can experience series of states as they become active or become standby.

Note: Packet processing only occurs on an active Oracle Enterprise Session Border Controller .

State	Description
Initial	When the Oracle Enterprise Session Border Controller is booting.
Becoming Active	When the Oracle Enterprise Session Border Controller has negotiated to become the active system, but is waiting the time that you set to become fully active. Packets cannot be processed in this state.
Active	When the Oracle Enterprise Session Border Controller is handling all media, signaling, and configuration processing.
Relinquishing Active	When the Oracle Enterprise Session Border Controller is giving up its Active status, but before it has become standby. This state is very brief.
Becoming Standby	When the Oracle Enterprise Session Border Controller is becoming the standby system but is waiting to become fully synchronized. It remains in this state for the period of time you set in the becoming-standby-time parameter, or until it is fully synchronized.
Standby	When the Oracle Enterprise Session Border Controller is fully synchronized with its active system in the HA node.
OutOfService	When the Oracle Enterprise Session Border Controller cannot become synchronized in the period of time you set in the becoming-standby-time parameter.

State Transition Sequences

When the active Oracle Enterprise Session Border Controller assumes its role as the as the active system, but then changes roles with the standby Oracle Enterprise Session Border Controller to become standby, it goes through the following sequence of state transitions:

- Active
- RelinquishingActive

- BecomingStandby
- Standby

  When the standby Oracle Enterprise Session Border Controller assumes its role as the standby system, but then changes roles with the active Oracle Enterprise Session Border Controller to become active, it goes through the following sequence of state transitions:
- Standby
- BecomingActive
- Active

HA Features

HA nodes support configuration checkpointing, which you are required to set up so that the configurations across the HA node are synchronized. In addition, you can set up the following optional HA node features:

- Multiple rear interface support
- Gateway link failure detection and polling

Multiple Rear Interfaces

Configuring your HA node to support multiple rear interfaces eliminates the possibility that either of the rear interfaces you configure for HA support will become a single point of failure. Using this feature, you can configure individual Oracle Enterprise Session Border Controller s with multiple destinations on the two rear interfaces, creating an added layer of failover support.

When you configure your HA node for multiple rear interface support, you can use last two rear interfaces (wancom1 and wancom2) for HA—the first (wancom0) being used for Oracle Enterprise Session Border Controller management. You can connect your Oracle Enterprise Session Border Controller s using any combination of wancom1 and wancom2 on both systems. Over these rear interfaces, the Oracle Enterprise Session Border Controller s in the HA node share the following information:

- Health
- Media flow
- Signaling
- Configuration

For example, if one of the rear interface cables is disconnected or if the interface connection fails for some other reason, all health, media flow, signaling, and configuration information can be checkpointed over the other interface.

Health information is checkpointed across all configured interfaces. However, media flow, signaling, and configuration information is checkpointed across one interface at a time, as determined by the Oracle Enterprise Session Border Controller ’s system HA processes.

Configuration Checkpointing

During configuration checkpointing, all configuration activity and changes on one Oracle Enterprise Session Border Controller are automatically mirrored on the other. Checkpointed transactions include adding, deleting, or modifying a configuration on the active Oracle Enterprise Session Border Controller . This means that you only need to perform configuration tasks on the active Oracle Enterprise Session Border Controller because the standby system will go through the checkpointing process and synchronize its configuration to reflect activity and changes.

Because of the way configuration checkpointing works, the ACLI save-config and activate-config commands can only be used on the active Oracle Enterprise Session Border Controller .

- When you use the ACLI save-config command on the active Oracle Enterprise Session Border Controller , the standby Oracle Enterprise Session Border Controller learns of the action and updates its own configuration. Then the standby Oracle Enterprise Session Border Controller saves the configuration automatically.
- When you use the ACLI activate-config command on the active Oracle Enterprise Session Border Controller , the standby Oracle Enterprise Session Border Controller learns of the action and activates its own, updated configuration.

The ACLI acquire-config command is used to copy configuration information from one Oracle Enterprise Session Border Controller to another.

Gateway Link Failure Detection and Polling

In an HA node, the Oracle Enterprise Session Border Controller s can poll for and detect media interface links to the gateways as they monitor ARP connectivity. The front gateway is assigned in the network interface configuration, and is where packets are forwarded out of the originator's LAN.

The Oracle Enterprise Session Border Controller monitors connectivity using ARP messages that it exchanges with the gateway. The Oracle Enterprise Session Border Controller sends regular ARP messages to the gateway in order to show that it is still in service; this is referred to as a heartbeat message. If the Oracle Enterprise Session Border Controller deems the gateway unreachable for any of the reasons discussed in this section, a network-level alarm is generated and an amount you configure for this fault is subtracted from the system's health score.

TheOracle Enterprise Session Border Controller generates a gateway unreachable network-level alarm if the Oracle Enterprise Session Border Controller has not received a message from the media interface gateway within the time you configure for a heartbeat timeout. In this case, The Oracle Enterprise Session Border Controller will send out ARP requests and wait for a reply. If no reply is received after resending the set number of ARP requests, the alarm remains until you clear it. The health score also stays at its reduced amount until you clear the alarm.

When valid ARP requests are once again received, the alarm is cleared and system health scores are increased the appropriate amount.

You can configure media interface detection and polling either on a global basis in the SD HA nodes/redundancy configuration or on individual basis for each network interface in the network interface configuration.

Note: To improve the detection of link failures, the switchport connected to the NIU should have Spanning Tree disabled. Enabling Spanning Tree stops the switchport from forwarding frames for several seconds after a reset. This prevents the NIU from reaching the gateway and generates a "gateway unreachable" network-level alarm.

Before Configuring a High Availability (HA) Pair

Note: When you configure an HA pair, you must use the same password for both Oracle Enterprise Session Border Controllers.

Before configuring the parameters that support HA, complete the following steps.

1. Establish the physical connections between the Oracle Enterprise Session Border Controllers. Avoid breaking the physical link (over the rear interfaces) between the Oracle Enterprise Session Border Controllers in an HA node. If the physical link between the Oracle Enterprise Session Border Controllers breaks, they will both attempt to become the active system and HA will not function as designed.
2. Confirm that both Oracle Enterprise Session Border Controllers are set to the same time. Use the ACLI show clock command to view the system time. If the Oracle Enterprise Session Border Controllers show different times, use the system-timeset command to change the time.

   Oracle recommends that you use NTP to synchronize your Oracle Enterprise Session Border Controllers, so that they have a common stratum time source.
3. HA nodes use ports 1 and 2 as the HA interfaces. Set port 0 on the rear panel of the Oracle Enterprise Session Border Controller chassis as the boot and management interface. You configure the rear interfaces during the physical interface configuration.
4. For ACLI configuration, you need to know the target names of the Oracle Enterprise Session Border Controllers making up the HA node. The target name of the system is reflected in the ACLI's system prompt. For example, in the ACMEPACKET# system prompt, ACMEPACKET is the target name.

   You can also see and set the target name in the Oracle Enterprise Session Border Controller boot parameters.

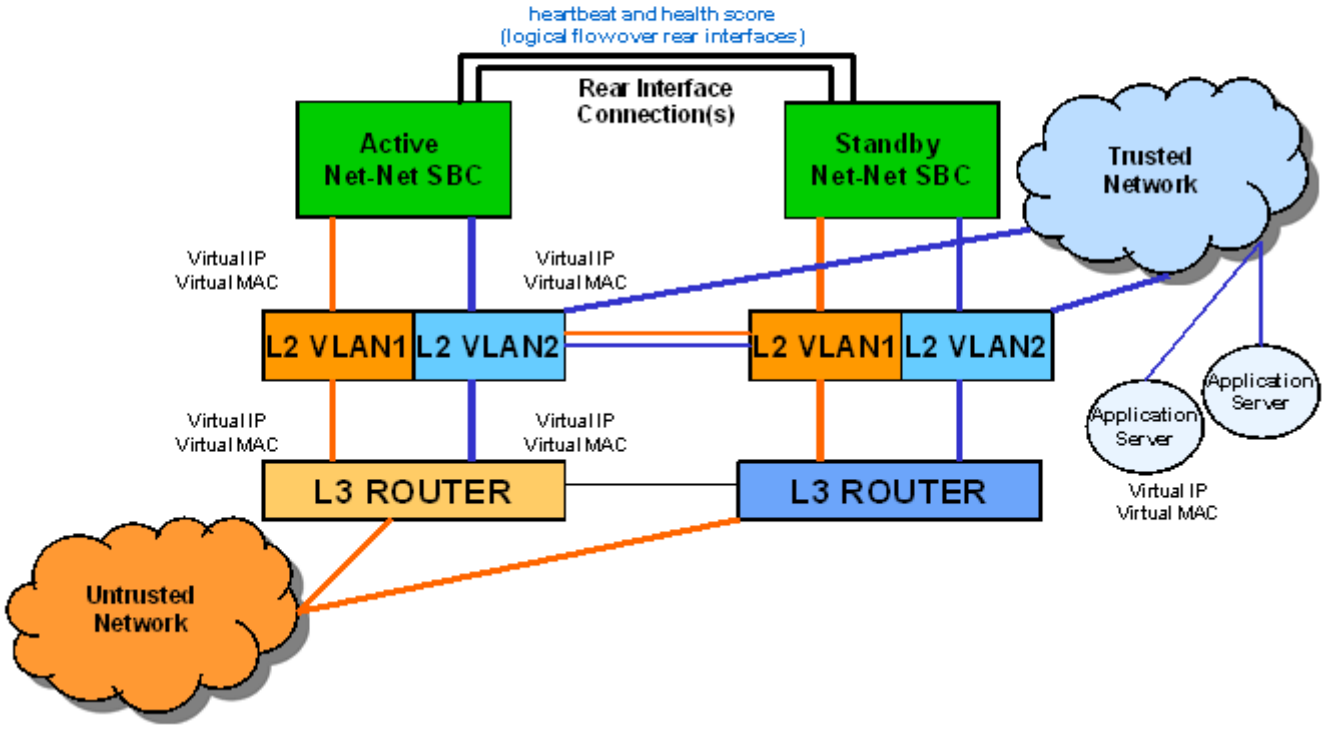
   Note: The target name is case sensitive.

5. Devise virtual MAC addresses so that, if a switchover happens, existing sessions are not interrupted.

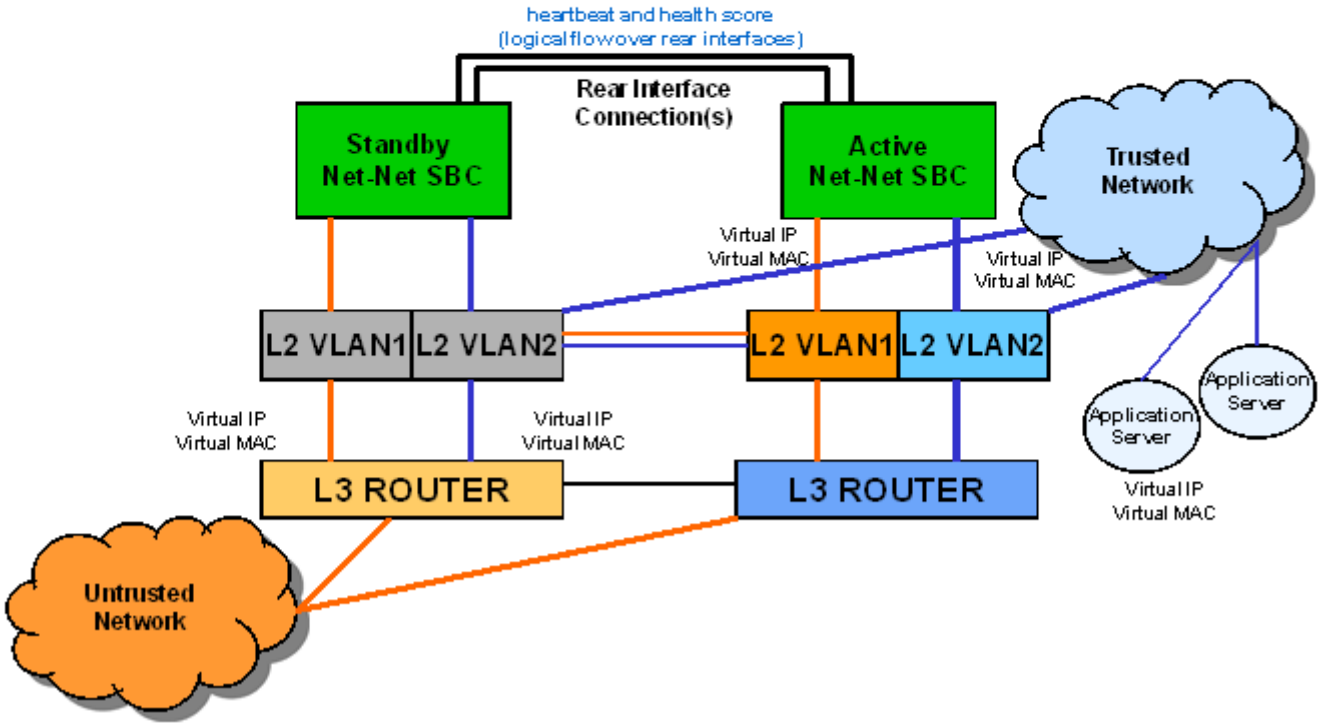
HA Node Connections

To use HA, you must establish Layer 2 and Layer 3 networks that interconnect two Oracle Enterprise Session Border Controllers and support HA with the required physical network connections. The basic network set-up in the following diagram shows an HA node deployment where each system is connected to its own Layer 2 switch. This set-up provides a measure of added redundancy in the event that one of the switches fails.

Here, the active system is using the virtual MAC and IP addresses.



In the second diagram, the same network is shown with the HA node having experienced a switchover. The previously standby Oracle Enterprise Session Border Controller has taken over the active role in the HA node and is using the virtual IP and MAC addresses.



Note: Switches should never be in master-slave mode. If they are, HA will not work correctly.

The following are hardware set-up and location considerations for placing an HA Node:

- You must set up each Oracle Enterprise Session Border Controller according to the requirements and safety precautions set out in the *Oracle Communications System Hardware Installation Guide*.
- Each Oracle Enterprise Session Border Controller's media interfaces must be connected to the same switches (or other network entities), as shown in the diagram above.
- The length of the shielded crossover 10/100 category 5 Ethernet cable that connects the Oracle Enterprise Session Border Controllers from the rear interfaces must be able to reach from the configured rear interface on one Oracle Enterprise Session Border Controller to the configured rear interface on the other.

HA nodes use Oraclerder element redundancy protocol for its tasks. This protocol uses a connection between the rear interfaces of two Oracle Enterprise Session Border Controllers to checkpoint the following information: health, state, media flow, signaling, and configuration.

We recommend that you use shielded category 5 (RJ45) crossover cables for all 10/100 Ethernet connections used for HA.

You can set up either single or multiple rear interface support for your HA node. For single interface support, one cable connects the two Oracle Enterprise Session Border Controllers; for multiple interface support, two cables are used. However, the software configurations for each type of connection mode are different.

When you make these connections, do not use port 0 (wancom0) on the rear interface of the Oracle Enterprise Session Border Controller chassis; that port should only be used for Oracle Enterprise Session Border Controller management. Instead, use ports 1 and 2 (wancom1 and wancom2).

To cable Oracle Enterprise Session Border Controllers using single rear interface support:

1. Using a 10/100 category 5 crossover cable, insert one end into either port 1 (wancom1) or port 2 (wancom2) on the rear interface of the first Oracle Enterprise Session Border Controller.
2. Insert the other end of the cable into port 1 or port 2 on the rear interface of the second Oracle Enterprise Session Border Controller. We recommend that you use corresponding ports on the two systems. That is, use port 1 on both systems or use port 2 on both systems.
3. Perform software configuration for these interfaces as described in this chapter.

   To cable Oracle Enterprise Session Border Controllers using multiple rear interface support:
4. Using a 10/100 category 5 crossover cable, insert one end into port 1 on the rear interface of the first Oracle Enterprise Session Border Controller.
5. Insert the other end of that cable into port 1 on the rear interface of the second Oracle Enterprise Session Border Controller to complete the first physical connection.
6. Using a second 10/100 category 5 cable, insert one end into port 2 on the rear interface of the first Oracle Enterprise Session Border Controller.
7. Insert the other end of this second cable in port 2 on the rear interface of the second Oracle Enterprise Session Border Controller to complete the second physical connection.
8. Perform software configuration for these interfaces as described in this chapter.

Virtual MAC Addresses

In order to create the HA node, you need to create virtual MAC addresses for the media interfaces. You enter these addresses in virtual MAC address parameters for physical interface configurations where the operation type for the interface is media.

The HA node uses shared virtual MAC (media access control) and virtual IP addresses for the media interfaces. When there is a switchover, the standby Oracle Enterprise Session Border Controller sends out an ARP message using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. Virtual MAC addresses are actually unused MAC addresses that based on the Oracle Enterprise Session Border Controller's root MAC address.

The MAC address is a hardware address that uniquely identifies each Oracle Enterprise Session Border Controller. Given that, the virtual MAC address you configure allows the HA node to appear as a single system from the perspective of other network devices. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted through the standby Oracle Enterprise Session Border Controller.

Depending on the type of physical layer cards you have installed, you can create MAC addresses as follows: Four Ethernet (MAC) address for each configured four-port GigE physical interface card.

Virtual MAC Address Configuration

To create a virtual MAC address:

1. Determine the Ethernet address of the Oracle Enterprise Session Border Controller by using the ACLI show interfaces command. This command only works if you have already set up physical interface configurations. Otherwise, you will get no output.

   The example below shows you where the Ethernet address information appears; this sample has been shortened for the sake of brevity. For each type of physical interface card, the Oracle Enterprise Session Border Controller displays the following:

```
ACMEPACKET# show interfaces
f00 (media slot 0, port 0)
     Flags: UP BROADCAST MULTICAST ARP RUNNING
     Type: GIGABIT_ETHERNET
```

```
    Admin State: enabled
    Auto Negotiation: enabled
    Internet address: 10.10.0.10     Vlan: 0
    Broadcast Address: 10.10.255.255
    Netmask: 0xffff0000
    Gateway: 10.10.0.1
    Ethernet address is 00:08:25:01:07:64
```

2. Identify the root portion of the Ethernet (MAC) address.

   Each Oracle Enterprise Session Border Controller has MAC addresses assigned to it according to the following format: 00:08:25:XX:YY:ZN where:

   - 00:08:25 refers to Acme Packet
   - XX:YY:ZN refers to the specific Oracle Enterprise Session Border Controller
   - N is a 0-f hexadecimal value available for the Oracle Enterprise Session Border Controller

   In this example, the root part of this address is 00:08:25:XX:YY:Z.

3. To create an unused MAC address (that you will use as the virtual MAC address) take the root MAC address you have just identified. Replace this N value with unused hexadecimal values for the Oracle Enterprise Session Border Controller: 8, 9, e, or f.

   In other words, you change the last digit of the MAC address to either 8, 9, e, or f depending on which of those address are not being used.

   For example, for an HA node with MAC address bases of 00:08:25:00:00:00 and 00:08:25:00:00:10, the following addresses would be available for use at virtual MAC addresses:

   - 00:08:25:00:00:08
   - 00:08:25:00:00:09
   - 00:08:25:00:00:0e
   - 00:08:25:00:00:0f
   - 00:08:25:00:00:18
   - 00:08:25:00:00:19
   - 00:08:25:00:00:1e
   - 00:08:25:00:00:1f

   Corresponding media interfaces in HA nodes must have the same virtual MAC addresses. Given that you have various physical interface card options, the following points illustrate how virtual MAC address can be shared:

   If you are using a four-port GigE physical interface card, both the active Oracle Enterprise Session Border Controllerand the standby Oracle Enterprise Session Border Controller might have the following virtual MAC address scheme for the slots:

   - Slot 0 _ 00:08:25:00:00:0e and 00:08:25:00:00:0f
   - Slot 1 - 00:08:25:00:00:1e and 00:08:25:00:00:1f

   Note: Note the virtual MAC addresses you have created so that you can reference them easily when you are configuring the physical interfaces for HA.

HA Node Connections

You can begin software configuration for your HA node after you have:

- Completed the steps for physical set-up and connection.
- Noted the target name of the Oracle Enterprise Session Border Controllers that make up the HA node.
- Configured the virtual MAC addresses that you need, according to the type of physical interface cards installed on your Oracle Enterprise Session Border Controller.

HA Node Connection Configuration

If you are using HA, you need to set the physical interface configuration parameters described in this section to establish successful connections. These parameters are for rear and media interfaces.

To access physical interface menu in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type phy-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)#
```

   From this point, you can configure physical interface parameters. To view all physical interfaces parameters, enter a ? at the system prompt.

Rear Interfaces

You can use port 1 (wancom1) or port 2 (wancom2) as interfaces to support HA. Do not use port 0 (wancom 0) as that port is reserved for carrying management traffic.

Make sure that the physical connections you have made on the rear panel of your Oracle Enterprise Session Border Controllers correspond to the configurations you enter for physical interfaces. You can connect Oracle Enterprise Session Border Controllers through multiple rear interfaces. For multiple rear interface connectivity, cable both port 1 and port 2 (wancom1 and wancom2) on one Oracle Enterprise Session Border Controller to port1 and port 2 on the other Oracle Enterprise Session Border Controller in the HA node.

The Oracle Enterprise Session Border Controller's HA function depends heavily on health scores to determine the active and standby roles in an HA node. You can set the amount that will be subtracted from a Oracle Enterprise Session Border Controller's health score in the event that a management interface fails for any reason. For example, a connection might become invalid or a cable might be removed inadvertently.

The following example shows how a configured physical interface will appear in the ACLI for an HA node:

```
phy-interface
        name                            wancom1
        operation-type                  Control
        port                            1
        slot                            0
        virtual-mac
        wancom-health-score             20
```

To establish rear interfaces for use in an HA node using the ACLI:

1. Access the physical interface menu.
2. name—Set a name for the interface using any combination of characters entered without spaces. For example: wancom1.
3. operation-type—Set this parameter to Control.
4. slot—Set this parameter to 0.
5. port—Set this parameter to 1 or 2.
6. wancom-health-score—Enter the number value between 0 and 100. This value will be subtracted from the Oracle Enterprise Session Border Controller's health score in the event that a rear interface link fails. We recommend that you change this value from its default (50), and set it to 20.

   This value you set here is compared to the active and emergency health score thresholds you establish in the Oracle Enterprise Session Border Controller HA node (redundancy) configuration.
7. For multiple rear interface support, configure the remaining, unused rear interfaces with the appropriate values.

The following example shows configuration for multiple rear interface support.

```
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# name wancom1
ACMEPACKET(phy-interface)# operation-type control
ACMEPACKET(phy-interface)# port 1
ACMEPACKET(phy-interface)# wancom-health-score 20
ACMEPACKET(phy-interface)# done
ACMEPACKET(phy-interface)# name wancom2
ACMEPACKET(phy-interface)# operation-type control
ACMEPACKET(phy-interface)# port 2
ACMEPACKET(phy-interface)# wancom-health-score 20
ACMEPACKET(phy-interface)# done
```

Media Interface Virtual MAC Addresses

To configure HA for the media interfaces in an HA node, you must set one or more virtual MAC addresses, according to the type of physical layer cards you have installed on your Oracle Enterprise Session Border Controller.

To set a virtual MAC address using the ACLI:

1. Access the physical interface configuration.
2. Configure all relevant parameters as noted in the Physical Interfaces section of this guide's *System Configuration* chapter.

   Since virtual MAC addresses are used for media interfaces only, verify that the operation type is set to media.
3. virtual-mac—Enter the virtual MAC address that you have created using the steps in the Virtual MAC Addresses section.

HA Node Parameters

To establish a pair of Oracle Enterprise Session Border Controllers as an HA node, you need to configure basic parameters that govern how the Oracle Enterprise Session Border Controllers:

- Transition on switchover
- Share media and call state information
- Checkpoint configuration data

The following example shows what an HA configuration might look like in the ACLI.

```
redundancy-config
        state                           enabled
        log-level                       WARNING
        health-threshold                75
        emergency-threshold             50
        port                            9090
        advertisement-time              500
        percent-drift                   210
        initial-time                    1250
        becoming-standby-time           45000
        becoming-active-time            100
```

You need to configure the two Oracle Enterprise Session Border Controllers to be HA node peers. To enable configuration checkpointing, you must to configure two peers in the ACLI, one for the primary and one for the secondary Oracle Enterprise Session Border Controller. The HA node peers configuration also allows you to configure destinations for where to send health and state information. Unless you create Oracle Enterprise Session Border Controller peers and destinations configurations, HA will not work properly.

The following example shows what an HA configuration might look like in the ACLI.

```
peer
                name                            netnetsd1
                state                           enabled
                type                            Primary
```

```
                destination
                        address                         169.254.1.1:9090
network-interface               wancom1:0
peer
                name                            netnetsd2
                state                           enabled
                type                            Secondary
                destination
                        address                         169.254.1.2:9090
                        network-interface               wancom1:0
```

HA Node Parameter Configuration

To configure general HA node parameters using the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type redundancy and press Enter.

   ```
   ACMEPACKET(system)# redundancy
   ```

   From here, you configure basic HA node parameters. To view all basic HA node parameters, enter a ? at the system prompt.

4. state—Leave this parameter set to enabled for HA to work. To stop HA operation, set this parameter to disabled. The default value is enabled. The valid values are:

   - enabled | disabled

5. log-level—Set the log level you want to use for the HA system process. The value you set in this field overrides any log level value you set for the entire Oracle Enterprise Session Border Controller in the system configuration process log level parameter. The default value is INFO which allows you to receive a moderate amount of detail. The valid values are:

   - emergency | critical | major | minor | warning | notice | info | trace | debug | detail

6. health-threshold—Enter a value between 0 and 100 to set the health score at which the Oracle Enterprise Session Border Controllers in the HA node gracefully exchange active-standby roles. The default value is 75. The valid range is:

   - Minimum—1
   - Maximum—100

   For example, if this field is set to 75 and the active Oracle Enterprise Session Border Controller's health score falls below that point, the standby Oracle Enterprise Session Border Controller will take over the active role. However, Oracle Enterprise Session Border Controller will only take over the active role if its own health score is 75 or better.

7. emergency-threshold—Enter the health score for the standby Oracle Enterprise Session Border Controller to become active immediately. The default value is 50. The valid range is:

   - Minimum—0
   - Maximum—100

   If the standby Oracle Enterprise Session Border Controller is initializing and the active Oracle Enterprise Session Border Controller's health score is below the health threshold, the standby Oracle Enterprise Session Border Controller will take the active role and there will be a graceful switchover. If the active Oracle Enterprise Session Border Controller's health score is below the emergency threshold, then the switchover will be immediate.

If the standby Oracle Enterprise Session Border Controller has a health score below the emergency threshold and the active Oracle Enterprise Session Border Controlleris unhealthy, the active Oracle Enterprise Session Border Controller will not give up its active role.

8. advertisement-time—Enter the number of milliseconds to set how often Oracle Enterprise Session Border Controllers in an HA node inform each other of their health scores.

   We recommend you leave this parameter set to it's default, 500. The valid range is:

   - Minimum—50
   - Maximum—999999999

9. percent-drift—Enter the percentage of the advertisement time that you want one member of the HA node to wait before considering the other member to be out of service. For the standby Oracle Enterprise Session Border Controller, this is the time it will wait before taking the active role in the HA node. The default value is 210. The valid range is:

   - Minimum—100
   - Maximum—65535

10. initial-time—Enter the number of milliseconds to set the longest amount of time the Oracle Enterprise Session Border Controller will wait at boot time to change its state from initial to either becoming active or becoming standby. The default value is 1250. The valid range is:

    - Minimum—5
    - Maximum—999999999

11. becoming-standby-time—Enter the number of milliseconds the Oracle Enterprise Session Border Controller waits before becoming standby, allowing time for synchronization. If it is not fully synchronized within this time, it will be declared out of service.

    We recommend that you do not set this parameter below 45000. If a large configuration is being processed, we recommend setting this parameter to 180000 to allow enough time for configuration checkpointing. The default value is 45000. The valid range is:

    - Minimum—5
    - Maximum—999999999

12. becoming-active-time—Enter the number of milliseconds that the standby Oracle Enterprise Session Border Controller takes to become active in the event that the active Oracle Enterprise Session Border Controller fails or has an intolerably decreased health score. The default value is 100. The valid range is:

    - Minimum—5
    - Maximum—999999999

HA Node Peer Configuration

To configure a Oracle Enterprise Session Border Controller as an HA node peer:

1. From the redundancy menu, type peers and press Enter.

```
ACMEPACKET(system)# redundancy
ACMEPACKET(redundancy)# peers
```

2. state—Enable or disable HA for this Oracle Enterprise Session Border Controller. The default value is enabled. The valid values are:

   - enabled | disabled

3. name—Set the name of the HA node peer as it appears in the target name boot parameter.

   This is also the name of your system that appears in the system prompt. For example, in the system prompt ACMEPACKET1#, ORACLE1 is the target name for that Oracle Enterprise Session Border Controller.

4. type—These values refer to the primary and secondary utility addresses in the network interface configuration. To determine what utility address to use for configuration checkpointing, set the type of Oracle Enterprise Session Border Controller: primary or secondary.

Note: You must change this field from unknown, its default. The valid values are:

- primary—Set this type if you want the Oracle Enterprise Session Border Controller to use the primary utility address.
- secondary—Set this type if you want the Oracle Enterprise Session Border Controller to use the secondary utility address.
- unknown—If you leave this parameter set to this default value, configuration checkpointing will not work.

HA Node Health And State Configuration

To configure where to send health and state information within an HA node:

1. From the peers configuration, type destinations and press Enter.

```
ACMEPACKET(rdncy-peer)# destinations
ACMEPACKET(rdncy-peer-dest)#
```

2. address—Set the destination IPv4 address and port of the other Oracle Enterprise Session Border Controller in the HA node to which this Oracle Enterprise Session Border Controller will send HA-related messages. This value is an IPv4 address and port combination that you enter as: IPAddress:Port. For example, 169.254.1.1:9090.
   - The IPv4 address portion of this value is the same as the IPv4 address parameter set in a network interface configuration of the other Oracle Enterprise Session Border Controller in the HA node.
   - The port portion of this value is the port you set in the Oracle Enterprise Session Border Controller HA Node/ redundancy configuration for the other Oracle Enterprise Session Border Controller in the node.
3. network-interface—Set the name and subport for the network interface where the Oracle Enterprise Session Border Controller receives HA-related messages. Valid names are wancom1 and wancom2. This name and subport combination must be entered as name:subport; for example, wancom1:0.

   The network interface specified in this parameter must be linked to a physical interface configured with rear interface parameters. The physical interface's operation type must be control or maintenance, and so the subport ID portion of this parameter is 0. The subport ID is the VLAN tag.

Synchronizing Configurations

You can synchronize the Oracle Enterprise Session Border Controllers (E-SBC) in your High Availability (HA) node in the following ways:

- Automatically — Set up configuration checkpointing within the HA node.
- Manually — Check whether or not configurations in the HA node are synchronized, and then copy configuration data from one E-SBC to the other.

When you initially configure a new HA node, copy the configuration data manually from one E-SBC to the other. When you complete the process, you can configure your HA node to automatically synchronize configurations.

Oracle recommends that you configure the HA node for configuration checkpointing because that is the most reliable way to ensure that both systems have the same configuration.

Synchronize HA Peers

The process for synchronizing the peers in a High Availability (HA) node for the first time by way of the ACLI includes the following steps.

1. Create a complete configuration on the active Oracle Enterprise Session Border Controller (E-SBC). Include all HA node parameters and all rear interface configurations. Confirm that the rear interfaces are configured to send and receive information across the HA node.
2. On the active E-SBC, save the configuration.
3. On the active E-SBC, reboot to run the new configuration.

Use the ACLI show health command to see that the active E-SBC booted without a peer. This changes after you copy the configuration to the standby E-SBC and activate the configuration.

4. On the standby E-SBC, perform the ACLI acquire-config command to copy the configuration from the active E-SBC. Use the acquire-config command with the IPv4 address of wancom 0 on the active E-SBC.

```
ACMEPACKET2# acquire-config 192.168.12.4
```

The IPv4 address of wancom 0 on the active E-SBC is the IPv4 address portion of the value displayed for the inet on ethernet boot parameter. The following codeblock shows an example of the inet on ethernet value that the system displays when you view the boot parameters:

```
inet on ethernet (e)     : 192.168.12.4:ffff0000
```

5. When the copying process (acquire-config) is complete, reboot the standby E-SBC to activate the configuration. The system boots and displays start-up information.
6. Confirm that the HA node synchronized the configurations by using the ACLI display-current-cfg-version and display-running-cfg-version commands:

```
ACMEPACKET# display-current-cfg-version
Current configuration version is 3
ACMEPACKET# display-running-cfg-version
Running configuration version is 3
ACMEPACKET# display-current-cfg-version
Current configuration version is 3
ACMEPACKET# display-running-cfg-version
Running configuration version is 3
```

In the preceding example, all configuration versions—current and running—are the same number (3).

Using Configuration Checkpointing

The Oracle Enterprise Session Border Controller's primary and secondary utility addresses support configuration checkpointing, allowing the standby Oracle Enterprise Session Border Controller to learn configuration changes from the active Oracle Enterprise Session Border Controller. This means that you only have to enter configuration changes on the active Oracle Enterprise Session Border Controller for the configurations across the HA node to be updated.

Configuration checkpointing uses parameters in the network interface and in the Oracle Enterprise Session Border Controller HA Nodes/redundancy configurations.

If you are using configuration checkpointing, you also need to set up two Oracle Enterprise Session Border Controller peer configurations: one the primary, and one for the secondary.

HA Configuration Checkpointing

You need to first set applicable network interface configuration parameters, and then establish applicable parameters in the Oracle Enterprise Session Border Controller HA node (redundancy) configuration.

We recommend that you do not change the configuration checkpointing parameters in the redundancy configuration. Using the defaults, this feature will function as designed.

Note: Remember to set the appropriate type parameter in the HA node redundancy peers configuration.

For the network interface, these parameters appear as they do in the following example when you use the ACLI. This example has been shortened for the sake of brevity.

```
pri-utility-addr              169.254.1.1
sec-utility-addr              169.254.1.2
```

For the Oracle Enterprise Session Border Controller HA node (redundancy) configuration, these parameters appear as they do in the following example when you use the ACLI. This example has been shortened for the sake of brevity. You should not change these values without consultation from Oracle Technical Support or your Oracle Systems Engineer.

```
cfg-port                      1987
cfg-max-trans                 10000
```

```
cfg-sync-start-time             5000
cfg-sync-comp-time              1000
```

To configure HA configuration checkpointing in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type network-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# network-interface
   ACMEPACKET(network-interface)#
   ```

   From here, you can configure network interface parameters. To view all network interfaces parameters, enter a ? at the system prompt.

4. pri-utility-addr—Enter the utility IP address for the primary HA peer in an HA architecture.

   This address can be any unused IP address within the subnet defined for the network interface. For example, given a network interface of with the IPv4 address 168.0.4.15/24 (identifying the host associated with the network interface), the possible range of unused IPv4 addresses is 168.0.4.1 to 168.0.4.254. Your network administrator will know which IPv4 addresses are available for use.

5. sec-utility-addr—Enter the utility IP address for the secondary Oracle Enterprise Session Border Controller peer in an HA architecture.

   Usually, this IP address is usually the next in the sequence up from the primary utility address. It is also generated from the range of unused IP addresses within the subnet defined for the network interface.

6. Save your work and exit the network interface configuration.

   ```
   ACMEPACKET(network-interface)# done
   ACMEPACKET(network-interface)# exit
   ACMEPACKET(system)#
   ```

7. Access the system HA node/redundancy configuration by typing redundancy at the system prompt and then press Enter.

   ```
   ACMEPACKET(system)# redundancy
   ACMEPACKET(redundancy)#
   ```

   Note: We strongly recommend that you keep the default settings for the parameters Steps 8 through 11.

8. cfg-port—Enter the port number for sending and receiving configuration checkpointing messages. Setting this to zero (0) disables configuration checkpointing. The default value is 1987. The valid values are:

   - Minimum—0, 1025
   - Maximum—65535

9. cfg-max-trans—Enter the number of HA configuration checkpointing transactions that you want to store. The active Oracle Enterprise Session Border Controller maintains the transaction list, which is acquired by the standby Oracle Enterprise Session Border Controller. Then the standby system uses the list to synchronize its configuration with active system. The default value is 10000. The valid range is:

   - Minimum—0
   - Maximum—4294967295

     Transactions include: modifications, additions, and deletions. If the maximum number of stored transactions is reached, the oldest transactions will be deleted as new transactions are added.

10. cfg-sync-start-time—Enter the number of milliseconds before the Oracle Enterprise Session Border Controller tries to synchronize by using configuration checkpointing. On the active Oracle Enterprise Session Border Controller, this timer is continually reset as the Oracle Enterprise Session Border Controller checks to see that it is still in the active role. If it becomes standby, it waits this amount of time before it tries to synchronize.

We recommend you leave this field at its default value, 5000, so that configuration checkpointing can function correctly. The valid range is:

- Minimum—0
- Maximum—4294967295

11. cfg-sync-comp-time—Enter the number of milliseconds that the standby Oracle Enterprise Session Border Controller waits before checkpointing to obtain configuration transaction information after the initial checkpointing process is complete.

We recommend you leave this field at its default value, 1000, so that configuration checkpointing can function correctly. The valid range is:

- Minimum—0
- Maximum—4294967295

12. Save your work and exit the redundancy configuration.

```
ACMEPACKET(redundancy)# done
ACMEPACKET(redundancy)# exit
ACMEPACKET(system)#
```

Manually Checking Configuration Synchronization

You can check that the current and active configurations are synchronized across the HA node. The current configuration is the one with which you are currently working, and the active configuration is the one active on the system.

To confirm that the systems in the HA node have synchronized configurations:

1. On the active Oracle Enterprise Session Border Controller in the Superuser menu, enter the following ALCI commands and press Enter. Note the configuration version numbers for comparison with those on the standby Oracle Enterprise Session Border Controller.

- display-current-cfg-version—Shows the version number of the configuration you are currently viewing (for editing, updating, etc.).

```
ACMEPACKET# display-current-cfg-version
Current configuration version is 30
```

- display-running-cfg-version—Shows the version number of the active configuration running on the Oracle Enterprise Session Border Controller.

```
ACMEPACKET# display-running-cfg-version
Running configuration version is 30
```

2. On the standby Oracle Enterprise Session Border Controller, enter the following ALCI commands and press Enter. Note the configuration version numbers for comparison with those on the active Oracle Enterprise Session Border Controller.

```
ACMEPACKET# display-current-cfg-version
Current configuration version is 30
ACMEPACKET# display-running-cfg-version
Running configuration version is 30
```

3. Compare the configuration numbers. If the version numbers on the active Oracle Enterprise Session Border Controller match those on the standby Oracle Enterprise Session Border Controller, then the systems are synchronized.

If the version numbers do not match, you need to synchronize the Oracle Enterprise Session Border Controllers. You can do so using the ACLI acquire-config command.

Media Interface Link Detection and Gateway Polling

You can use media interface link detection and gateway polling globally on the Oracle Enterprise Session Border Controller, or you can override those global parameters on a per-network-interface basis.

- Use the Oracle Enterprise Session Border Controller HA node (redundancy) configuration to establish global parameters. When configured globally, they will appear like this in the ACLI:

```
gateway-heartbeat-interval     0
gateway-heartbeat-retry        0
gateway-heartbeat-timeout      1
gateway-heartbeat-health       0
```

- Use the network interface's gateway heartbeat configuration to override global parameters on a per-network-interface basis. When configured for the network interface, these parameters will appear like this in the ACLI:

```
gw-heartbeat
                state                          enabled
                heartbeat                      0
                retry-count                    0
                retry-timeout                  1
                health-score                   0
```

Media Interface Link Detection and Gateway Polling Configuration

To configure global media interface link detection and gateway polling:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type redundancy and press Enter.

```
ACMEPACKET(system)# redundancy
```

   From here, you can configure gateway heartbeat parameters. To view all gateway heartbeat parameters, enter a ? at the system prompt.

4. gateway-heartbeat-interval—Enter the number of seconds between heartbeats for the media interface gateway. Heartbeats are sent at this interval as long as the media interface is viable. The default value is 0. The valid range is:

   - Minimum—0
   - Maximum—65535

5. gateway-heartbeat-retry—Enter the number of heartbeat retries (subsequent ARP requests) to send to the media interface gateway before it is considered unreachable. The default value is 0. The valid range is:

   - Minimum—0
   - Maximum—65535

6. gateway-heartbeat-timeout—Enter the heartbeat retry time-out value in seconds. The default value is 1. The valid range is:

   - Minimum—0
   - Maximum—65535

     This parameter sets the amount of time between Oracle Enterprise Session Border Controller ARP requests to establish media interface gateway communication after a media interface gateway failure.

7. gateway-heartbeat-health—Enter the amount to subtract from the Oracle Enterprise Session Border Controller's health score if a media interface gateway heartbeat fails. If the value you set in the gateway time-out retry field is exceeded, this amount will be subtracted from the system's overall health score. The default value is 0. The valid range is:

   - Minimum—0
   - Maximum—100

Media Interface Link Detection and Gateway Polling Configuration 2

To configure media interface link detection and gateway polling on a per-network-interface basis in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# system
```

3. Type network-interface and press Enter.

```
ACMEPACKET(system)# network-interface
```

4. Type gw-heartbeat and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(network-interface)# gw-heartbeat
ACMEPACKET(gw-heartbeat)#
```

   From here, you can configure gateway heartbeat parameters for the network interface. To view all gateway heartbeat parameters, enter a ? at the system prompt.

5. state—Enable or disable the gateway heartbeat feature. The default value is enabled. The valid values are:

   - enabled | disabled

6. heartbeat—Enter the number of seconds between heartbeats for the media interface gateway. Heartbeats are sent at this interval as long as the media interface is viable. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—65535

     The value you configure in this field overrides any globally applicable value set in the gateway heartbeat interval parameter in the Oracle Enterprise Session Border Controller HA node (redundancy) configuration.

7. retry-count—Enter the number of heartbeat retries that you want sent to the media interface gateway before it is considered unreachable. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—65535

8. retry-timeout—Enter the heartbeat retry time-out value in seconds. The default value is 1. The valid range is:

   - Minimum—1
   - Maximum—65535

     This parameter sets the amount of time between system ARP requests to establish media interface gateway communication after a media interface gateway failure.

9. health-score—Enter the amount to subtract from the system's health score if a media interface gateway heartbeat fails; this parameter defaults to 0. If the value you set in the retry-time-out field is exceeded, this amount will be subtracted from the system's overall health score. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—100

Signaling Checkpointing

You can configure your HA node to checkpoint signaling for SIP and MGCP.

SIP Signaling Checkpointing

In the SIP configuration, you can set parameters that enable SIP signaling checkpointing across an HA node.

When configured, these parameters will appear in the ACLI as they do in example below.

Note: This example shows the default values being used, and we recommend that you do not change these values from their defaults.

```
red-sip-port                    1988
red-max-trans                   10000
red-sync-start-time             5000
red-sync-comp-time              1000
```

Signaling Checkpointing Configuration

To configure SIP signaling checkpointing across an HA node in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-router and press Enter.

   ```
   ACMEPACKET(session-router)# sip-config
   ```

   From here, you can configure SIP parameters for HA nodes. To view all SIP configuration parameters, enter a ? at the system prompt.

   When configuring SIP for HA, you only need to set the parameters addressed in this procedure.

4. red-sip-port—Enter the port on which SIP signaling checkpointing messages are sent and received. The default value is 1988. A value of 0 disables the SIP signaling checkpointing. The valid range is:

   - Minimum—0, 1024
   - Maximum—65535

5. red-max-trans—Enter the maximum size of the transaction list, or how many SIP transactions you want to store in memory at one time. Oldest transactions will be discarded first in the event that the limit is reached. The default value is 10000. The valid range is:

   - Minimum—0
   - Maximum—999999999

6. red-sync-start-time—Enter the number of milliseconds before the Oracle Enterprise Session Border Controller will try to synchronize its signaling state checkpointing.

   If the active Oracle Enterprise Session Border Controller is still adequately healthy, this timer will simply reset itself. If for any reason the active Oracle Enterprise Session Border Controller has become the standby, it will start to checkpoint with the newly active system when this timer expires.

   We recommend that you leave this parameter set to its default, 5000. The valid range is:

   - Minimum—0
   - Maximum—999999999

7. red-sync-comp-time—Enter the number of milliseconds representing how frequently the standby Oracle Enterprise Session Border Controller checkpointing with the active Oracle Enterprise Session Border Controller to obtain the latest SIP signaling information. The first interval occurs after initial synchronizations of the systems.

   We recommend that you leave this parameter set to its default, 1000. The valid range is:

   - Minimum—0
   - Maximum—999999999

Media State Checkpointing

By default, the Oracle Enterprise Session Border Controller performs media checkpointing across the HA node for all signaling protocols. You can keep the default port set for redundancy media flows.

H.323 media high availability is supported through a TCP socket keep-alive, which determines whether or not the other end of a TCP/IP network connection is still in fact connected. This type of checkpointing prevents the listening side of a connection from waiting indefinitely when a TCP connection is lost. When there is a switchover in the HA node, the system that has just become active takes over sending TCP keep-alives. Media continues to flow until the session ends or the flow guard timers expire.

This parameter will appear in the ACLI as follows:

```
    red-flow-port                   1985
```

Media State Checkpointing Configuration

To configure media state checkpointing across an HA node in the ACLI:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type media-manager-config and press Enter.

   ```
   ACMEPACKET(media-manager)# media-manager-config
   ```

4. red-flow-port—Enter the port number for checkpointing media flows associated with the HA interface. This is the port where media flow checkpoint message are sent and received.

   Setting this field to 0 disables media state checkpointing. The default value is 1985. The valid range is:

   - Minimum—0, 1025
   - Maximum—65535

HA Media Interface Keepalive

In an HA node, it is possible for the two systems in the node to lose communication via the management (rear, wancom) interfaces. For example, wancom 1 and wancom 2 might become disconnected, and cause the heartbeat synchronization to fail. This type of failure causes communication errors because both systems try to assume the active role and thereby access resources reserved for the active system.

To avoid these types of conditions, you can enable an option instructing the standby system to take additional time before going to the active state. This check occurs through the system's media interfaces. Using it, the standby can determine whether or not there has been a true active failure.

In cases when the standby determines the active system has not truly failed, it will go out of service because it will have determined it no longer has up-to-date data from its active counterpart. You can restore functionality by re-establishing management (rear) interface communication between the system in the node, and then re-synchronizes the standby by rebooting it.

When you enable the media interface keepalive, the standby system in the HA node sends ARP requests to determine if the media interfaces' virtual IP address are active. There are two possible outcomes:

- If it receives responses to its ARP requests, the standby takes itself out of service—to prevent a conflict with the active.
- If it does not receive responses to its ARP requests within a timeout value you set, then standby assumes the active role in the HA node.

Impact to Boot-Up Behavior

With the HA media interface keepalive enabled, the Oracle Enterprise Session Border Controller might be in the initial state longer than if the feature were disabled because it requires more information about the media (front) interfaces.

HA Media Interface Keepalive Configuration

You turn the HA media interface keepalive on by setting a timeout value for the standby to receive responses to its ARP requests before it assumes the active role in the HA node. Keeping this parameter set to 0, its default, disables the keepalive

To enable the HA media interface keepalive:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type redundancy and press Enter.

```
ACMEPACKET(session-router)# redundancy
ACMEPACKET(redundancy)#
```

   If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. media-if-peercheck-time—Enter the amount of time in milliseconds for the standby system in an HA node to receive responses to its ARP requests via the media interface before it takes over the active role from its counterpart.

   The default is 0, which turns the HA media interface keepalive off. The maximum value is 500 milliseconds.

5. Save and activate your configuration.

RTC Notes

Starting in Release 4.1, the HA configuration is supported for real-time configuration (RTC). However, not all of the HA-related parameters are covered by RTC because of the impact on service it would cause to reconfigure these parameters dynamically.

This section sets out what parameters you should not dynamically reconfigure, or should dynamically reconfigure with care.

HA

Changes to the following ACLI parameters will have the noted consequences when dynamically reconfigured:

- cfg-max-trans—Changing this value could cause the activation time to lengthen slightly
- init-time, becoming-standby-time, and becoming-active-time—Changes take place only if the system is not transitioning between these states; otherwise the system waits until the transition is complete to make changes
- percent-drift and advertisement-time—Changes are communicated between nodes in the HA pair as part of regular health advertisements

In addition, the following parameters are not part of the RTC enhancement, for the reason specified in the right-hand column.

Parameter	Impact
state	Disrupts service
port	Disrupts service; leaves systems in an HA node without a means of communicating with each other
cfg-port	Disrupts service; leaves systems in an HA node without a means of communicating with each other

Parameter	Impact
cfg-max-trans	Disrupts service
cfg-sync-start-time	Disrupts configuration replication
cfg-sync-comp-time	Disrupts configuration replication

Protocol-Specific Parameters and RTC

In addition, you should not change any of the parameters related to HA that are part of protocol or media management configurations that are used for protocol/media checkpointing. These are:

- SIP configuration
  - red-max-trans
  - red-sync-start-time
  - red-sync-comp-time
- MGCP Configuration
  - red-mgcp-port
  - red-max-trans
  - red-sync-start-time
  - red-sync-comp-time
- Media Manager configuration
  - red-flow-port
  - red-mgcp-port
  - red-max-trans
  - red-sync-start-time
  - red-sync-comp-time

19

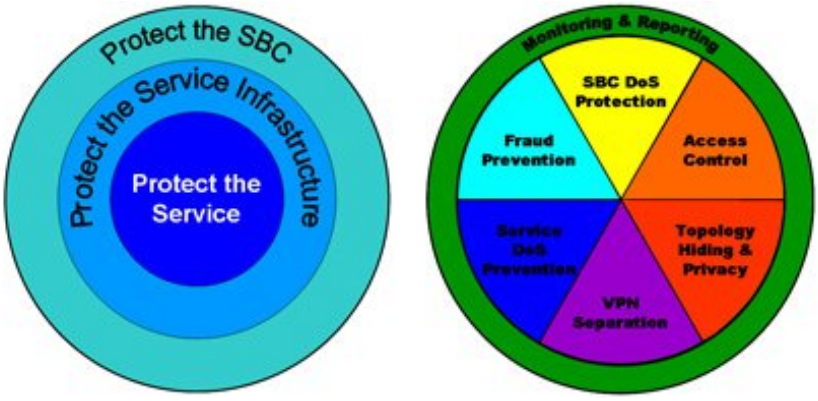
Security

This chapter explains Oracle Enterprise Session Border Controller security, which is designed to provide security for administrative security, VoIP and other multimedia services. It includes Admin Security, access control, DoS attack, and overload protection, which help secure service and protect the network infrastructure (including the Oracle Enterprise Session Border Controller). In addition, Oracle Enterprise Session Border Controller security lets legitimate users still place calls during attack conditions; protecting the service itself.

Security Overview

Oracle Enterprise Session Border Controller security includes the Net-SAFE framework's numerous features and architecture designs. Net-SAFE is a requirements framework for the components required to provide protection for the Session Border Controller (SBC), the service provider's infrastructure equipment (proxies, gateways, call agents, application servers, and so on), and the service itself.

The following diagrams illustrate Net-SAFE:



Each of Net-SAFE's seven functions consists of a collection of more specific features:

- Session border controller DoS protection: autonomic, SBC self-protection against malicious and non-malicious DoS attacks and overloads at Layers 2 to 4 (TCP, SYN, ICMP, fragments, and so on) and Layers 5 to 7 (SIP signaling floods, malformed messages, and so on).
- Access control: session-aware access control for signaling and media using static and dynamic permit/deny access control lists (ACLs) at layer 3 and 5.

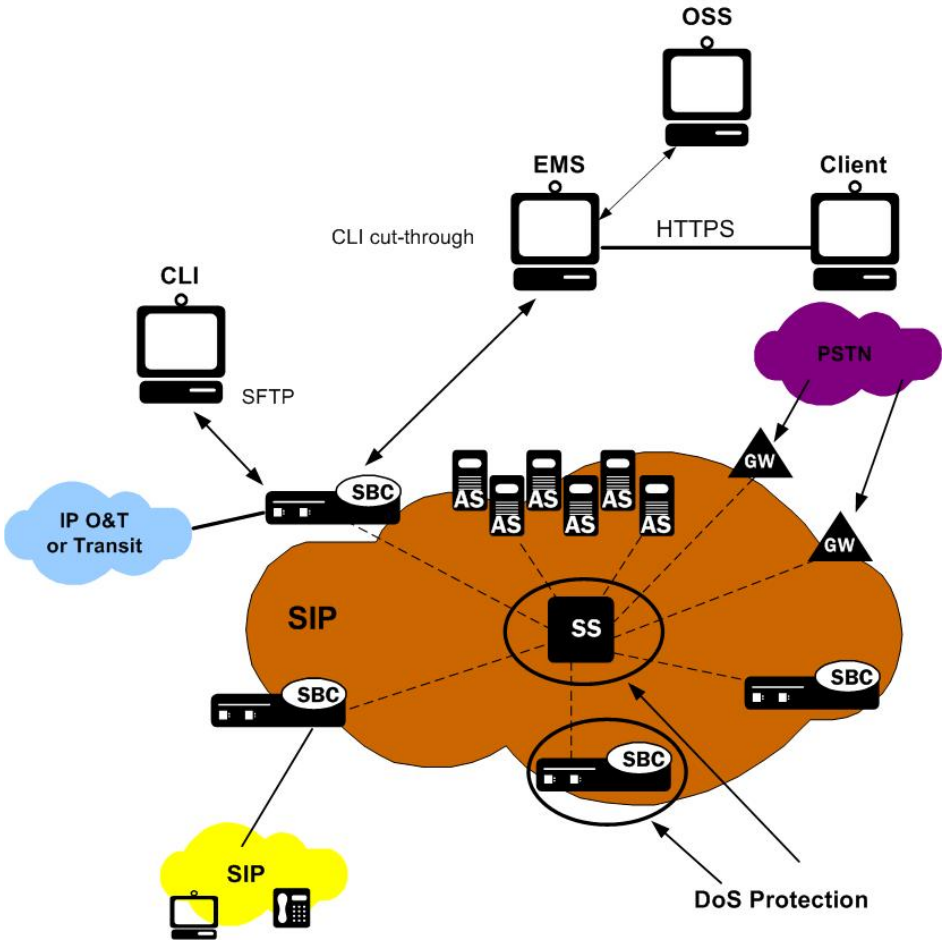
- Topology hiding and privacy: complete infrastructure topology hiding at all protocol layers for confidentiality and attack prevention security. Also, modification, removal or insertion of call signaling application headers and fields. Includes support for the SIP Privacy RFC.
- VPN separation: support for Virtual Private Networks (VPNs) with full inter-VPN topology hiding and separation, ability to create separate signaling and media-only VPNs, and with optional intra-VPN media hair-pinning to monitor calls within a VPN.
- Service infrastructure DoS prevention: per-device signaling and media overload control, with deep packet inspection and call rate control to prevent DoS attacks from reaching service infrastructure such as SIP servers, softswitches, application servers, media servers or media gateways.
- Fraud prevention: session-based authentication, authorization, and contract enforcement for signaling and media; and service theft protection.
- Monitoring and reporting: audit trails, event logs, access violation logs and traps, management access command recording, Call Detail Records (CDRs) with media performance monitoring, raw packet capture ability and lawful intercept capability. The monitoring method itself is also secured, through the use of SSH and SFTP, and through the ability to use a separate physical Ethernet port for management access.

Denial of Service Protection

This section explains the Denial of Service (DoS) protection for the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller DoS protection functionality protects softswitches and gateways with overload protection, dynamic and static access control, and trusted device classification and separation at Layers 3-5. The Oracle Enterprise Session Border Controller itself is protected from signaling and media overload, but more importantly the feature allows legitimate, trusted devices to continue receiving service even during an attack. DoS protection prevents the Oracle Enterprise Session Border Controller host processor from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Volume-based attack (flood) of valid or invalid call requests, signaling messages, and so on.
- Overload of valid or invalid call requests from legitimate, trusted sources

The following diagram illustrates DoS protection applied to the softswitch and to the Oracle Enterprise Session Border Controller.



Levels of DoS Protection

The multi-level Oracle Enterprise Session Border Controller DoS protection consists of the following strategies:

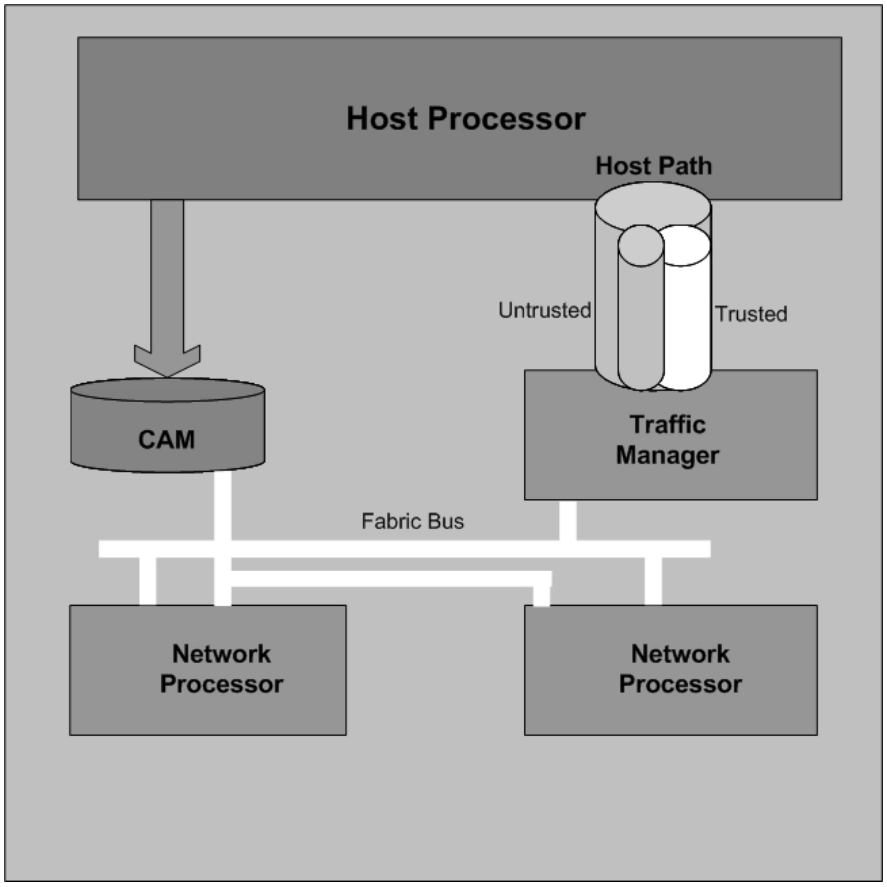
- Fast path filtering/access control: access control for signaling packets destined for the Oracle Enterprise Session Border Controller host processor as well as media (RTP) packets. The Oracle Enterprise Session Border Controller performs media filtering by using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted list and a deny list for each Oracle Enterprise Session Border Controller realm (although the actual devices can be dynamically trusted or denied by the Oracle Enterprise Session Border Controller based on configuration). You do not have to provision every endpoint/device on the Oracle Enterprise Session Border Controller, but instead retain the default values.
- Host path protection: includes flow classification, host path policing and unique signaling flow policing. Fast path filtering alone cannot protect the Oracle Enterprise Session Border Controller host processor from being overwhelmed by a malicious attack from a trusted source. The host path and individual signaling flows must be policed to ensure that a volume-based attack will not overwhelm the Oracle Enterprise Session Border Controller's normal call processing; and subsequently not overwhelm systems beyond it.

  The Oracle Enterprise Session Border Controller must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent. At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Oracle Enterprise Session Border Controller maintains two host paths, one for each class of traffic (trusted and untrusted), with different policing characteristics to ensure that fully trusted traffic always gets precedence.
- Host-based malicious source detection and isolation – dynamic deny list. Malicious sources can be automatically detected in real-time and denied in the fast path to block them from reaching the host processor.

About the Process

DoS attacks are handled in the Oracle Enterprise Session Border Controller's host path. The Oracle Enterprise Session Border Controller uses NAT table entries to filter out undesirable IP addresses; creating a deny list. After a packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager subsystem based on the sender's IP address. NAT table entries distinguish signaling packets coming in from different sources for policing purposes. The maximum number of policed calls that the Oracle Enterprise Session Border Controller can support is 16K (on 32K CAM / IDT CAM).

The Traffic Manager has two pipes, trusted and untrusted, for the signaling path. Each signaling packet destined for the host CPU traverses one of these two pipes.



Trusted Path

Packets from trusted devices travel through the trusted pipe in their own individual queues. In the Trusted path, each trusted device flow has its own individual queue (or pipe). The Oracle Enterprise Session Border Controller can dynamically add device flows to the trusted list by promoting them from the Untrusted path based on behavior; or they can be statically provisioned.

Trusted traffic is put into its own queue and defined as a device flow based on the following:

- source IP address
- source UDP/TCP port number
- destination IP address
- destination UDP/TCP port (SIP or MGCP interface to which it is sending)
- realm it belongs to, which inherits the Ethernet interface and VLAN it came in on

For example, SIP packets coming from 10.1.2.3 with UDP port 1234 to the Oracle Enterprise Session Border Controller SIP interface address 11.9.8.7 port 5060, on VLAN 3 of Ethernet interface 0:1, are in a separate Trusted

queue and policed independently from SIP packets coming from 10.1.2.3 with UDP port 3456 to the same Oracle Enterprise Session Border Controller address, port and interface.

Data in this flow is policed according to the configured parameters for the specific device flow, if statically provisioned. Alternatively, the realm to which endpoints belong have a default policing value that every device flow will use. The defaults configured in the realm mean each device flow gets its own queue using the policing values. As shown in the previous example, if both device flows are from the same realm and the realm is configured to have an average rate limit of 10K bytes per second (10KBps), each device flow will have its own 10KBps queue. They are not aggregated into a 10KBps queue.

The individual flow queues and policing lets the Oracle Enterprise Session Border Controller provide each trusted device its own share of the signaling, separate the device's traffic from other trusted and untrusted traffic, and police its traffic so that it can't attack or overload the Oracle Enterprise Session Border Controller (therefore it is trusted, but not completely).

Address Resolution Protocol Flow

The Address Resolution Protocol (ARP) packets are given their own trusted flow with the bandwidth limitation of 8 Kbps. ARP packets are able to flow smoothly, even when a DoS attack is occurring.

Untrusted Path

Packets (fragmented and unfragmented) that are not part of the trusted or denied list travel through the untrusted pipe. In the untrusted path, traffic from each user/device goes into one of 2048 queues with other untrusted traffic. Packets from a single device flow always use the same queue of the 2048 untrusted queues, and 1/2048th of the untrusted population also uses that same queue. To prevent one untrusted endpoint from using all the pipe's bandwidth, the 2048 flows defined within the path are scheduled in a fair-access method. As soon as the Oracle Enterprise Session Border Controller decides the device flow is legitimate, it will promote it to its own trusted queue.

All 2048 untrusted queues have dynamic sizing ability, which allows one untrusted queue to grow in size, as long as other untrusted queues are not being used proportionally as much. This dynamic queue sizing allows one queue to use more than average when it is available. For example, in the case where one device flow represents a PBX or some other larger volume device. If the overall amount of untrusted packets grows too large, the queue sizes rebalance, so that a flood attack or DoS attack does not create excessive delay for other untrusted devices.

In the usual attack situations, the signaling processor detects the attack and dynamically demotes the device to denied in the hardware by adding it to the deny ACL list. Even if the Oracle Enterprise Session Border Controller does not detect an attack, the untrusted path gets serviced by the signaling processor in a fair access mechanism. An attack by an untrusted device will only impact 1/1000th of the overall population of untrusted devices, in the worst case. Even then there's a probability of users in the same 1/1000th percentile getting in and getting promoted to trusted.

IP Fragment Packet Flow

All fragment packets are sent through their own 1024 untrusted flows in the Traffic Manager. The first ten bits (LSB) of the source address are used to determine which fragment-flow the packet belongs to. These 1024 fragment flows share untrusted bandwidth with already existing untrusted-flows. In total, there are 2049 untrusted flows: 1024-non-fragment flows, 1024 fragment flows, and 1 control flow.

Fragmented ICMP packets are qualified as ICMP packets rather than fragment packets. Fragment and non-fragmented ICMP packets follow the trusted-ICMP-flow in the Traffic Manager, with a bandwidth limit of 8Kbs.

Fragment Packet Loss Prevention

You can set the maximum amount of bandwidth (in the max-untrusted-signaling parameter) you want to use for untrusted packets. However, because untrusted and fragment packets share the same amount of bandwidth for policing, any flood of untrusted packets can cause the Oracle Enterprise Session Border Controller to drop fragment packets.

To prevent fragment packet loss, you can set the fragment-msg-bandwidth. When it is set to any value other than 0 (which disables it), the Oracle Enterprise Session Border Controller:

- Provides for a separate policing queue for fragment packets (separate from that used for untrusted packets)

- Uses this new queue to prevent fragment packet loss when there is a flood from untrusted endpoints.

When you set up a queue for fragment packets, untrusted packets likewise have their own queue—meaning also that the max-untrusted-signaling and min-untrusted-signaling values are applied to the untrusted queue.

Static and Dynamic ACL Entry Limits

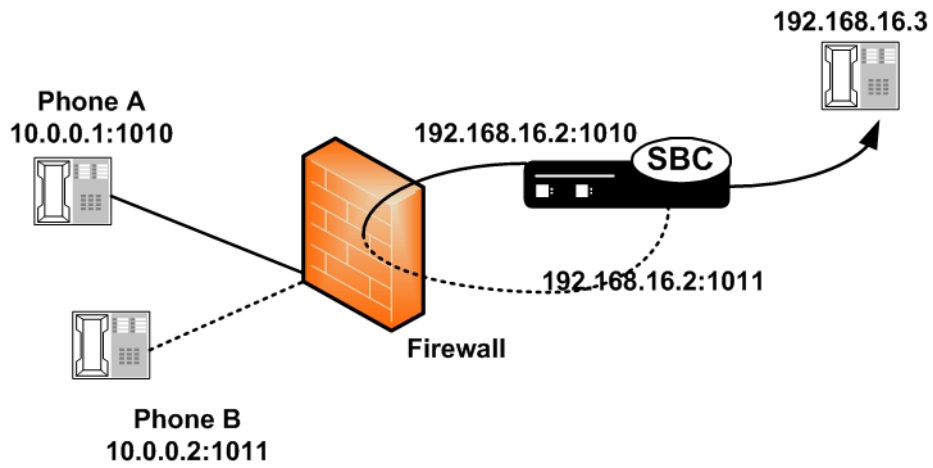
The Oracle Enterprise Session Border Controller can simultaneously police a maximum of 250,000 trusted device flows, while at the same time denying an additional 32,000 attackers. If list space becomes full and additional device flows need to be added, the oldest entries in the list are removed and the new device flows are added.

Dynamic Deny for HNT

Dynamic deny for HNT has been implemented on the Oracle Enterprise Session Border Controller for cases when callers are behind a NAT or firewall. Without this feature, if one caller behind a NAT or firewall were denied, the Oracle Enterprise Session Border Controller would also deny all other users behind the same NAT or firewall. This would be true even for endpoints behind the firewall that had not crossed threshold limits you set for their realm; all endpoints behind the firewall would go out of service. In the following diagram, both Phone A and Phone B would be denied because their IP addresses would be translated by the firewall to the same IPv4 address (192.168.16.2).

However, dynamic deny for HNT allows the Oracle Enterprise Session Border Controller to determine, based on the UDP/TCP port, which endpoints should be denied and which should be allowed. The Oracle Enterprise Session Border Controller can determine that even though multiple endpoints originating behind a firewall appear with the same IPv4 address, those addresses use different ports and are unique.

As shown in the diagram below, the ports from Phone A and Phone B remain unchanged. This way, if Phone A violates the thresholds you have configured, the Oracle Enterprise Session Border Controller can block traffic from Phone A while still accepting traffic from Phone B.



Host and Media Path Protection Process

The Oracle Enterprise Session Border Controller Network Processors (NPs) check the deny and permit lists for received packets, and classify them as trusted, untrusted or denied (discard). Only packets to signaling ports and dynamically signaled media ports are permitted. All other packets sent to Oracle Enterprise Session Border Controller ports are filtered. Only packets from trusted and untrusted (unknown) sources are permitted; any packet from a denied source is dropped by the NP hardware. The Traffic Manager manages bandwidth policing for trusted and untrusted traffic, as described earlier. Malicious traffic is detected in the host processor and the offending device is dynamically added to denied list, which enables early discard by the NP. Devices become trusted based on behavior detected by the Signaling Processor, and dynamically added to the trusted list. This process enables the proper classification by the NP hardware. All other traffic is untrusted (unknown).

Session Director Access Control

You an create static trusted/untrusted/deny lists with source IP addresses or IP address prefixes, UDP/TDP port number or ranges, and based on the appropriate signaling protocols. Furthermore, the Oracle Enterprise Session Border Controller can dynamically promote and demote device flows based on the behavior, and thus dynamically creates trusted, untrusted, and denied list entries.

Access Control for Hosts

ACLs are supported for all VoIP signaling protocols on the Oracle Enterprise Session Border Controller: SIP, H.323, and MGCP. The Oracle Enterprise Session Border Controller loads ACLs so they are applied when signaling ports are loaded. The following rules apply to static NAT entries based on your configuration:

- If there are no ACLs applied to a realm that have the same configured trust level as that realm, the Oracle Enterprise Session Border Controller adds a default NAT entry using the realm parameters.
- If you configure a realm with none as its trust level and you have configured ACLs, the Oracle Enterprise Session Border Controller only applies the ACLs.
- If you set a trust level for the ACL that is lower than the one you set for the realm, the Oracle Enterprise Session Border Controller will not add a separate NAT entry for the ACL.

ACLs provide access control based on destination addresses when you configure destination addresses as a way to filter traffic. You can set up a list of access control exceptions based on the source or the destination of the traffic.

For dynamic ACLs based on the promotion and demotion of endpoints, the rules of the matching ACL are applied.

Access Control Endpoint Classification Capacity and DoS

The following capacities are for both IPv4 and IPv6 endpoints.

Platform	Denied	Trusted	Media	Untrusted	Dynamic Trusted	ARP	VLAN
AP3820	32000	8000	8000	2000	250000	4000	4000
AP4500	32000	8000	32000	2000	250000	4000	4000

Media Access Control

The media access control consists of media path protection and pinholes through the firewall. Only RTP and RTCP packets from ports dynamically negotiated through signaling (SIP, H.323, MGCP) are allowed, which reduces the chance of RTP hijacking. Media access depends on both the destination and source RTP/RTCP UDP port numbers being correct, for both sides of the call.

Host Path Traffic Management

The host path traffic management consists of the dual host paths discussed earlier:

- Trusted path is for traffic classified by the system as trusted. You can initially define trusted traffic by ACLs, as well as by dynamically promoting it through successful SIP or MGCP registration, or a successful call establishment. You can configure specific policing parameters per ACL, as well as define default policing values for dynamically-classified flows. Traffic for each trusted device flow is limited from exceeding the configured values in hardware. Even an attack from a trusted, or spoofed trusted, device cannot impact the system.
- Untrusted path is the default for all unknown traffic that has not been statically provisioned otherwise. For example, traffic from unregistered endpoints. Pre-configured bandwidth policing for all hosts in the untrusted path occurs on a per-queue and aggregate basis.

Traffic Promotion

Traffic is promoted from untrusted to trusted list when the following occurs:

- successful SIP registration for SIP endpoints

- successful RSIP response for MGCP endpoints
- successful session establishment for SIP or MGCP calls

Malicious Source Blocking

Malicious source blocking consists of monitoring the following metrics for each source:

- SIP transaction rate (messages per second)
- SIP call rate (call attempts per second)
- Nonconformance/invalid signaling packet rate

Device flows that exceed the configured invalid signaling threshold, or the configured valid signaling threshold, within the configured time period are demoted, either from trusted to untrusted, or from untrusted to denied classification.

Blocking Actions

Blocking actions include the following:

- Dynamic deny entry added, which can be viewed through the ACLI.
- SNMP trap generated, identifying the malicious source

Dynamically added deny entries expire and are promoted back to untrusted after a configured default deny period time. You can also manually clear a dynamically added entry from the denied list using the ACLI.

Protecting Against Session Agent Overloads

You can prevent session agent overloads with registrations by specifying the registrations per second that can be sent to a session agent.

ARP Flood Protection Enhancements

Enhancements have been made to the way the Oracle Enterprise Session Border Controller provides ARP flood protection. In releases prior to Release C5.0, there is one queue for both ARP requests and responses, which the Oracle Enterprise Session Border Controller polices at a non-configurable limit (eight kilobytes per second). This method of ARP protection can cause problems during an ARP flood, however. For instance, gateway heartbeats the Oracle Enterprise Session Border Controller uses to verify (via ARP) reachability for default and secondary gateways could be throttled; the Oracle Enterprise Session Border Controller would then deem the router or the path to it unreachable, decrement the system's health score accordingly. Another example is when local routers send ARP requests for the Oracle Enterprise Session Border Controller's address are throttled in the queue; the Oracle Enterprise Session Border Controller never receives the request and so never responds, risking service outage.

The solution implemented to resolve this issue is to divide the ARP queue in two, resulting in one ARP queue for requests and a second for responses. This way, the gateway heartbeat is protected because ARP responses can no longer be flooded from beyond the local subnet. In addition, the Oracle Enterprise Session Border Controllers in HA nodes generate gateway heartbeats using their shared virtual MAC address for the virtual interface.

In addition, this solution implements a configurable ARP queue policing rate so that you are not committed to the eight kilobytes per second used as the default in prior releases. The previous default is not sufficient for some subnets, and higher settings resolve the issue with local routers sending ARP request to the Oracle Enterprise Session Border Controller that never reach it or receive a response.

As a security measure, in order to mitigate the effect of the ARP table reaching its capacity, configuring the media-manager option, active-arp, is advised. Enabling this option causes all ARP entries to get refreshed every 20 minutes.

Dynamic Demotion for NAT Devices

In addition to the various ways the Oracle Enterprise Session Border Controller already allows you to promote and demote devices to protect itself and other network elements from DoS attacks, it can now block off an entire NAT device. The Oracle Enterprise Session Border Controller can detect when a configurable number of devices behind a NAT have been blocked off, and then shut off the entire NAT's access.

This dynamic demotion of NAT devices can be enabled for an access control (ACL) configuration or for a realm configuration. When you enable the feature, the Oracle Enterprise Session Border Controller tracks the number of endpoints behind a single NAT that have been labeled untrusted. It shuts off the NAT's access when the number reaches the limit you set.

The demoted NAT device then remains on the untrusted list for the length of the time you set in the deny-period.

DDoS Protection from Devices Behind a NAT

A DDoS attack could be crafted such that multiple devices from behind a single NAT could overwhelm the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller would not detect this as a DDoS attack because each endpoint would have the same source IP but multiple source ports. Because the Oracle Enterprise Session Border Controller allocates a different CAM entry for each source IP:Port combination, this attack will not be detected. This feature remedies such a possibility.

Configuring DoS Security

This section explains how to configure the Oracle Enterprise Session Border Controller for DoS protection.

Configuration Overview

Configuring Oracle Enterprise Session Border Controller DoS protection includes masking source IP and port parameters to include more than one match and configuring guaranteed minimum bandwidth for trusted and untrusted signaling path. You can also configure signaling path policing parameters for individual source addresses. Policing parameters are defined as peak data rate (in bytes/sec), average data rate (in bytes/sec), and maximum burst size.

You can configure deny list rules based on the following:

- ingress realm
- source IP address
- source port
- transport protocol (TCP/UDP)
- application protocol (SIP, MGCP, H.323)

Changing the Default Oracle Enterprise Session Border Controller Behavior

The Oracle Enterprise Session Border Controller automatically creates permit untrusted ACLs that let all sources (address prefix of 0.0.0.0/0) reach each configured realm's signaling interfaces, regardless of the realm's address prefix. To deny sources or classify them as trusted, you create static or dynamic ACLs, and the global permit untrusted ACL to specifically deny sources or classify them as trusted. Doing this creates a default permit-all policy with specific deny and permit ACLs based on the realm address prefix.

You can change that behavior by configuring static ACLs for realms with the same source prefix as the realm's address prefix; and with the trust level set to the same value as the realm. Doing this prevents the permit untrusted ACLs from being installed. You then have a default deny all ACL policy with specific static permit ACLs to allow packets into the system.

Example 1 Limiting Access to a Specific Address Prefix Range

The following example shows how to install a permit untrusted ACL of source 12.34.0.0/16 for each signalling interface/port of a realm called access. Only packets from within the source address prefix range 12.34.0.0/16, destined for the signaling interfaces/port of the realm named access, are allowed. The packets go into untrusted queues until they are dynamically demoted or promoted based on their behavior. All other packets are denied/dropped.

- Configure a realm called access and set the trust level to low and the address prefix to 12.34.0.0/16.

- Configure a static ACL with a source prefix of 12.34.0.0/16 with the trust level set to low for the realm named access.

Example 2 Classifying the Packets as Trusted

Building on Example 1, this example shows how to classify all packets from 12.34.0.0/16 to the realm signaling interfaces as trusted and place them in a trusted queue. All other packets from outside the prefix range destined to the realm's signaling interfaces are allowed and classified as untrusted; then promoted or demoted based on behavior.

You do this by adding a global permit untrusted ACL (source 0.0.0.0) for each signaling interface/port of the access realm. You configure a static ACL with a source prefix 12.34.0.0/16 and set the trust level to high.

Adding this ACL causes the Oracle Enterprise Session Border Controller to also add a permit trusted ACL with a source prefix of 12.34.0.0/16 for each signaling interface/port of the access realm. This ACL is added because the trust level of the ACL you just added is high and the realm's trust level is set to low. The trust levels must match to remove the global permit trusted ACL.

Example 3 Installing Only Static ACLs

This example shows you how to prevent the Oracle Enterprise Session Border Controller from installing the global permit (0.0.0.0) untrusted ACL.

- Configure a realm with a trust level of none.
- Configure static ACLs for that realm with the same source address prefix as the realm's address prefix, and set the trust level to any value.

The system installs only the static ACLs you configure.

Access Control List Configuration

To configure access control lists:

1. Access the access-control configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# access-control
ACMEPACKET(access-control)#
```

2. Type select to choose and configure an existing object.

```
ACMEPACKET(access-control)# select
<src-ip>:
1: src 0.0.0.0; 0.0.0.0; realm01; ; ALL
```

3. realm-id—Enter the ID of the host's ingress realm.
4. source-address—Enter the source IPv4 address and port number for the host in the following format:

```
<IP address>[/number of address bits>][:<port>][/<port bits>]
```

   For example:

```
10.0.0.1/24:5000/14
10.0.0.1/16
10.0.0.1/24:5000
10.0.0.1:5000
```

   You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. The default value is 0.0.0.0.

5. destination-address—(Is ignored if you configure an application protocol in step 7.) Enter the destination IPv4 address and port for the destination in the following format:

```
<IP address>[/number of address bits>][:<port>[/<port bits>]]
```

You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. The default value is 0.0.0.0.

6. application-protocol—Enter the application protocol type for this ACL entry. The valid values are:

   - SIP | H.323 | MGCP | None

   Note: If application-protocol is set to none, the destination-address and port will be used. Ensure that your destination-address is set to a non-default value (0.0.0.0.)

7. transport-protocol—Select the transport-layer protocol configured for this ACL entry. The default value is ALL. The valid values are:

   - ALL | TCP | UDP

8. access—Enter the access control type or trusted list based on the trust-level parameter configuration for this host. The default value is permit. The valid values are:

   - permit—Puts the entry into the untrusted list. The entry is promoted or demoted according to the trust level set for this host.
   - deny—Puts the entry in the deny list.

9. average-rate-limit—Indicate the sustained rate in bytes per second for host path traffic from a trusted source within the realm. The default value is 0. A value of 0 means policing is disabled. The valid range is:

   - Minimum—0
   - Maximum—999999999

10. trust-level—Indicate the trust level for the host with the realm. The default value is none. The valid values are:

    - none—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
    - low—Host can be promoted to the trusted list or demoted to the deny list.
    - medium—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
    - high—Host is always trusted.

11. invalid-signal-threshold— Enter the number of invalid signaling messages that trigger host demotion. The value you enter here is only valid when the trust level is low or medium. Available values are:

    - Minimum—Zero (0) is disabled.
    - Maximum—999999999

    If the number of invalid messages exceeds this value based on the tolerance window parameter, configured in the media manager, the host is demoted.

    The tolerance window default is 30 seconds. Bear in mind, however, that the system uses the same calculation it uses for specifying "recent" statistics in show commands to determine when the number of signaling messages exceeds this threshold. This calculation specifies a consistent start time for each time period to compensate for the fact that the event time, such as a user running a show command, almost never falls on a time-period's border. This provides more consistent periods of time for measuring event counts.

    The result is that this invalid signal count increments for two tolerance windows, 60 seconds by default, within which the system monitors whether or not to demote the host. The signal count for the current tolerance window is always added to the signal count of the previous tolerance window and compared against your setting.

12. maximum-signal-threshold—Set the maximum number of signaling messages the host can send within the tolerance window. The value you enter here is only valid when the trust level is low or medium. The default value is 0, disabling this parameter. The valid range is:

    - Minimum—0
    - Maximum—999999999

    If the number of messages received exceeds this value within the tolerance window, the host is demoted.

13. untrusted-signal-threshold—Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is 0, disabling this parameter. The valid range is:

    - Minimum—0
    - Maximum—999999999

14. deny-period—Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is 30. The valid range is:

    - Minimum—0
    - Maximum—999999999

15. nat-trust-threshold—Enter the number of endpoints behind a NAT that must be denied for the Oracle Enterprise Session Border Controller to demote the NAT device itself to denied (dynamic demotion of NAT devices). The default is 0, meaning dynamic demotion of NAT devices is disabled. The range is from 0 to 65535.

    The following example shows access control configured for a host in the external realm.

```
access-control
        realm-id                        external
        source-address                  192.168.200.215
        destination-address             192.168.10.2:5000
        application-protocol            SIP
        transport-protocol              ALL
        access                          permit
        average-rate-limit              3343
        trust-level                     low
        invalid-signal-threshold        5454
        maximum-signal-threshold        0
        untrusted-signal-threshold      0
        deny-period                     0
```

    The following example of how to configure a black-list entry:

```
access-control
        realm-id                        external
        source-address                  192.168.200.200
        destination-address             192.168.10.2:5000
        application-protocol            SIP
        transport-protocol              ALL
        access                          deny
        average-rate-limit              0
        trust-level                     none
        invalid-signal-threshold        0
        maximum-signal-threshold        0
        untrusted-signal-threshold      0
        deny-period                     0
```

Host Access Policing

You can configure the Oracle Enterprise Session Border Controller to police the overall bandwidth of the host path.

To configure host access policing:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type media-manager and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. max-signaling-bandwidth—Set the maximum overall bandwidth available for the host path in bytes per second, which includes signaling messages from trusted and untrusted sources. It also includes any Telnet and FTP traffic on media ports. The default value is 1000000. The valid range is:
   - Minimum—71000
   - Maximum—10000000
5. max-untrusted-signaling—Set the percentage of the maximum signaling bandwidth you want to make available for messages coming from untrusted sources. This bandwidth is only available when not being used by trusted sources. The default value is 100 The valid range is:
   - Minimum—1
   - Maximum—100
6. min-untrusted-signaling—Set the percentage of the maximum signaling bandwidth you want reserved for the untrusted sources. The rest of the bandwidth is available for trusted resources, but can also be used for untrusted sources (see max-untrusted-signaling). The default value is 30 The valid range is:
   - Minimum—1
   - Maximum—100
7. fragment-msg-bandwidth—Enter the amount of bandwidth to use for the fragment packet queue. If you leave this parameter set to 0, then the Oracle Enterprise Session Border Controller will use the same queue for and share bandwidth between untrusted packets and fragment packets. The default value is zero (0). The valid range is:
   - Minimum—0
   - Maximum—10000000
8. tolerance-window—Set the size of the window used to measure host access limits. The value entered here is used to measure the invalid message rate and maximum message rate for the realm configuration. The default value is 30. The valid range is:
   - Minimum—0
   - Maximum—999999999

   The following example shows a host access policing configuration.

```
media-manager
        state                         enabled
        latching                      enabled
        flow-time-limit               86400
        initial-guard-timer           300
        subsq-guard-timer             300
        tcp-flow-time-limit           86400
        tcp-initial-guard-timer       300
        tcp-subsq-guard-timer         300
        tcp-number-of-ports-per-flow  2
        hnt-rtcp                      disabled
        algd-log-level                WARNING
        mbcd-log-level                WARNING
        home-realm-id
        red-flow-port                 1985
        red-mgcp-port                 1986
        red-max-trans                 10000
        red-sync-start-time           5000
        red-sync-comp-time            1000
        max-signaling-bandwidth       1000000
        max-untrusted-signaling       50
        min-untrusted-signaling       30
        tolerance-window              30
        rtcp-rate-limit               0
```

Configuring ARP Flood Protection

You do not need to configure the Oracle Enterprise Session Border Controller to enable the use of two separate ARP queues; that feature is enabled automatically.

If you want to configure the ARP queue policing rate, you can do so in the media manager configuration.

Note: this feature is not RTC-supported, and you must reboot your Oracle Enterprise Session Border Controller in order for your configuration changes to take effect.

To set the ARP queue policing rate:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Enter media-manager and press <Enter:.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. arp-msg-bandwidth—Enter the rate at which you want the Oracle Enterprise Session Border Controller to police the ARP queue; the value you enter is the bandwidth limitation in bytes per second. The default value is 32000. The valid range is:

   - Minimum—2000
   - Maximum—200000

5. Save your configuration.
6. Reboot your Oracle Enterprise Session Border Controller.

Access Control for a Realm

Each host within a realm can be policed based on average rate, peak rate, and maximum burst size of signaling messages. These parameters take effect only when the host is trusted. You can also set the trust level for the host within the realm. All untrusted hosts share the bandwidth defined for the media manager: maximum untrusted bandwidth and minimum untrusted bandwidth.

To configure access control for a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. addr-prefix—Set the IP address prefix used to determine if an IP address is associated with the realm. This value is then associated with the ACLs you create to determine packet access. The default value is 0.0.0.0.
5. average-rate-limit—Set the sustained rate for host path traffic from a trusted source within the realm in bytes per second. The default value is zero (0), disabling this parameter. The valid range is:

   - Minimum—0
   - Maximum—4294967295

6. access-control-trust-level—Set the trust level for the host within the realm. The default value is none. The valid values are:

- none—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
- low—Host can be promoted to the trusted list or demoted to the deny list.
- medium—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
- high—Host is always trusted.

7. invalid-signal-threshold— Enter the number of invalid signaling messages that trigger host demotion. The value you enter here is only valid when the trust level is low or medium. Available values are:

   - Minimum—Zero (0) is disabled.
   - Maximum—999999999

   If the number of invalid messages exceeds this value based on the tolerance window parameter, configured in the media manager, the host is demoted.

   The tolerance window default is 30 seconds. Bear in mind, however, that the system uses the same calculation it uses for specifying "recent" statistics in show commands to determine when the number of signaling messages exceeds this threshold. This calculation specifies a consistent start time for each time period to compensate for the fact that the event time, such as a user running a show command, almost never falls on a time-period's border. This provides more consistent periods of time for measuring event counts.

   The result is that this invalid signal count increments for two tolerance windows, 60 seconds by default, within which the system monitors whether or not to demote the host. The signal count for the current tolerance window is always added to the signal count of the previous tolerance window and compared against your setting.

8. maximum-signal-threshold—Set the maximum number of signaling messages one host can send within the window of tolerance. The host is demoted if the number of messages received by the Oracle Enterprise Session Border Controller exceeds the number set here. Valid only when the trust level is set to low or medium. The default value is zero (0), disabling this parameter. The valid range is:

   - Minimum—0
   - Maximum—4294967295

9. untrusted-signal-threshold—Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is zero (0), disabling the parameter. The valid range is:

   - Minimum—0
   - Maximum—4294967295

10. deny-period—Set the length of time an entry is posted on the deny list. The host is deleted from the deny lost after this time period. The default value is 30. A value of 0 disables the parameter. The valid range is:

    - Minimum—0
    - Maximum—4294967295

11. nat-trust-threshold—Enter the number of endpoints behind a NAT that must be denied for the Oracle Enterprise Session Border Controller to demote the NAT device itself to denied (dynamic demotion of NAT devices). The default is 0, meaning dynamic demotion of NAT devices is disabled. The range is from 0 to 65535.

    The following example shows a host access policing configuration.

```
realm-config
        identifier                      private
        addr-prefix                     192.168.200.0/24
        network-interfaces
                                        prviate:0
        mm-in-realm                     disabled
        mm-in-network                   enabled
        msm-release                     disabled
        qos-enable                      disabled
        max-bandwidth                   0
        ext-policy-svr
```

```
        max-latency                     0
        max-jitter                      0
        max-packet-loss                 0
        observ-window-size              0
        parent-realm
        dns-realm
        media-policy
        in-translationid
        out-translationid
        class-profile
        average-rate-limit              8000
        access-control-trust-level      medium
        invalid-signal-threshold        200
        maximum-signal-threshold        0
        untrusted-signal-threshold      500
        deny-period                     30
        symmetric-latching              disabled
        pai-strip                       disabled
        trunk-context
```

Configuring Overload Protection for Session Agents

The Oracle Enterprise Session Border Controller offers two methods to control SIP registrations to smooth the registration flow.

You can limit the:

- number of new register requests sent to a session agent (using the max-register-sustain-rate parameter)
- burstiness which can be associated with SIP registrations

The first method guards against the Oracle Enterprise Session Border Controller's becoming overwhelmed with register requests, while the second method guards against a transient registration that can require more than available registration resources.

SIP registration burst rate control allows you to configure two new parameters per SIP session agent—one that controls the registration burst rate to limit the number of new registration requests, and a second to set the time window for that burst rate. When the registration rate exceeds the burst rate you set, the Oracle Enterprise Session Border Controller responds to new registration requests with 503 Service Unavailable messages.

Note that this constraint is not applied to re-registers resulting from a 401 Unauthorized challenge request.

To configure overload protection for session agents:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. constraints—Enable this parameter to set the sustained rate window constraint you configure in the next step. The default value is disabled. The valid values are:

   - enabled | disabled

5. sustain-rate-window—Enter a number to set the sustained window period (in milliseconds) that is used to measure the sustained rate. The default value is zero (0). The valid range is:

   - Minimum—10
   - Maximum—4294967295

The value you set here must be higher than or equal to the value you set for the burst rate window.

Note: If you are going to use this parameter, you must set it to a minimum value of 10.

6. max-register-sustain-rate—Enter a number to set the maximum number of registrations per second you want sent to the session agent. The default value is zero (0), disabling the parameter. The valid range is:
   - Minimum—0
   - Maximum—4294967295
7. register-burst-window—Define the window size in seconds for the maximum number of allowable SIP registrations. 0 is the minimum and default value for this parameter; the maximum value is 999999999.
8. max-register-burst-rate—Enter the maximum number of new registrations you want this session agent to accept within the registration burst rate window. If this threshold is exceeded, the Oracle Enterprise Session Border Controller will respond to new registration requests with 503 Service Unavailable messages. 0 is the minimum and default value for this parameter; the maximum value is 999999999.
9. Save and activate your configuration.

Media Policing

Media policing controls the throughput of individual media flows in the Oracle Enterprise Session Border Controller, which in turn provides security and bandwidth management functionality. The media policing feature works for SIP, H.323, SIP-H.323, and MGCP/NCS protocols. The media policing feature also lets you police static flows and RTCP flows.

The term media policing refers to flows that go through the Oracle Enterprise Session Border Controller. Flows that are directed to the host application are not affected by media policing.

You can use media policing to protect against two potential security threats that can be directed against your Oracle Enterprise Session Border Controller:

- Media DoS—Once media flows are established through the Oracle Enterprise Session Border Controller, network resources are open to RTP media flooding. You can eliminate the threat of a media DoS attack by constraining media flows to absolute bandwidth thresholds.
- Bandwidth Piracy—Bandwidth policing ensures that sessions consume no more bandwidth than what is signaled for.

Policing Methods

The Oracle Enterprise Session Border Controller polices real-time traffic by using Constant Bit Rate (CBR) media policing. CBR policing is used when a media flow requires a static amount of bandwidth to be available during its lifetime. CBR policing best supports real-time applications that have tightly constrained delay variation. For example, voice and video streaming are prime candidates for CBR policing.

Session Media Flow Policing

Session media encompasses RTP and RTCP flows. In order to select policing constraints for these flows, the Oracle Enterprise Session Border Controller watches for the codec specified in an SDP or H.245 message. When a match is made between the codec listed in an incoming session request and a configured media-profile configuration element, the Oracle Enterprise Session Border Controller applies that media-profile's bandwidth policing constraint to the media flow about to start.

If multiple codecs are listed in the SDP message, the Oracle Enterprise Session Border Controller will use the media-profile with the most permissive media policing constraints for all of the flows associated with the session. If a codec in the H.245/SDP message is not found in any configured media-profile, the Oracle Enterprise Session Border Controller uses the media-profile with the most permissive media policing constraints configured. If no media-profiles are configured, there will be no session media flow policing.

If a mid-call change occurs, bandwidth policing is renegotiated.

Static Flow Policing

Static flows can also be policed in the same way as media flows are policed. A static flow configuration redirects flows entering the Oracle Enterprise Session Border Controller on a media interface. The redirection is based on realm, source, destination, and protocol. When a flow matches the configured static flow criteria, besides being redirected toward a specified destination, its rate can also be controlled based on a static flow policing parameter found in the static-flow element. Static flow policing operates obliviously to the data contained within the flow.

Configuration Notes

Review the following information before configuring your Oracle Enterprise Session Border Controller to perform media policing.

Session Media Flow Policing

Session media flow policing applies to both RTP and RTCP flows. Setting either of the parameters listed below to 0 disables media policing, letting RTP or RTCP flows pass through the system unrestricted.

- RTP Policing
  - Set in the media-profile configuration element's average-rate-limit parameter to police RTP traffic with the CBR policing method.
  - average-rate-limit—Establishes the maximum speed for a flow in bytes per second.
- RTCP Policing
  - Set in the media-manager-config configuration element's rtcp-rate-limit parameter to police RTCP traffic with the CBR policing method.
  - rtcp-rate-limit—Establishes the maximum speed for an RTCP flow in bytes per second.

Static Flow Policing

Static flow policing is configured with one parameter found in the static-flow configuration element. To configure CBR, you have to set the average-rate-limit parameter to a non-zero value. Setting the parameter listed below to 0 disables static flow policing, effectively letting the flow pass through the Oracle Enterprise Session Border Controller unrestricted.

In a CBR configuration, the average-rate-limit parameter determines the maximum bandwidth available to the flow.

- average-rate-limit—Establishes the maximum speed for a static flow in bytes per second.

Note: Static flow policing is not necessarily tied to any type of media traffic, it can affect flows of any traffic type.

Media Policing Configuration for RTP Flows

You can configure media policing in the media-profile configuration element using the ACLI. In the following example, you will configure media policing for the G723 media profile.

To configure media policing for RTP flows:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type media-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# media-profile
   ```

4. Select an existing media profile to which you will add policing constraints.

   ```
   ACMEPACKET(media-profile)# select
   <name>:
   ```

```
1: audio   4=G723     RTP/AVP 16 0 0 0
selection:1
ACMEPACKET(media-profile)#
```

From this point, you can configure media policing parameters. To view all media-profile parameters, enter a ? at the system prompt

5. average-rate-limit—Enter the maximum rate in bytes per second for any flows that this media-profile polices. The default value is zero (0), disabling media policing. The valid range is:

   - Minimum—0
   - Maximum—125000000

   Average rate limit values for common codecs:
   - PCMU—80000 Bps
   - G729—26000 Bps

   The following example shows a media-profile configuration element configured for media policing.

```
media-profile
        name                            G723
        media-type                      audio
        payload-type                    4
        transport                       RTP/AVP
        req-bandwidth                   16
        frames-per-packet               0
        parameters
        average-rate-limit              15000
```

Media Policing Configuration for RTCP Flows

You can configure media policing for RTCP flows by using the ACLI.

To configure media policing for RTCP flows:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type media-manager and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. rtcp-rate-limit—Enter the RTCP policing constraint in bytes per second. The default value is zero (0). The valid range is:

   - Minimum—0
   - Maximum—125000000

Media Policing Configuration for Static Flows

You can configure media policing for static flows using the ACLI.

To configure media policing for static flows:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media-manager path.

```
ACMEPACKET(configure)# media-manager
```

3. Type static-flow and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# static-flow
ACMEPACKET(static-flow)#
```

4. Select an existing static flow to which you will add policing constraints.

```
ACMEPACKET(static-flow)# select
<in-dest-ip>:
1: dest 0.0.0.0; src 192.168.2.1/24; static-flow-in-realm; UDP
selection:1
```

From this point, you can configure media policing parameters for static flows. To view all static-flow parameters, enter a ? at the system prompt

5. average-rate-limit—Enter the maximum rate in bytes per second for any flows that this static-flow polices. The default value is zero (0). The valid range is:

- Minimum—0
- Maximum—125000000

The following example shows a static-flow configuration element configured for media policing.

```
static-flow
        in-realm-id                     static-flow-in-realm
        in-source                       192.168.2.1/24
        in-destination                  0.0.0.0
        out-realm-id                    static-flow-out-realm
        out-source                      192.168.128.1/24
        out-destination                 0.0.0.0
protocol                        UDP
        average-rate-limit              15000
```

RTP Payload Type Mapping

The Oracle Enterprise Session Border Controller maintains a default list of RTP payload types mapped to textual encoding names as defined in RFC 3551.

The following table defines the preconfigured payload type for standard encodings.

Payload Type	Encoding Name	Audio (A) / Video (V)	Clock Rate
0	PCMU	A	8000
4	G723	A	8000
8	PCMA	A	8000
9	G722	A	8000
15	G728	A	8000
18	G729	A	8000

If you configure any payload type to encoding name mappings, the default mappings will be ignored. You must then manually enter all payload type mappings you use in the media-profile configuration element.

ITU-T to IANA Codec Mapping

The Oracle Enterprise Session Border Controller maintains a list of ITU-T (H.245) codecs that map to IANA RTP codecs. An ITU codec is directly mapped to an IANA Encoding Name for media profile lookups. All codecs are normalized to IANA codec names before any matches are made. New ITU-T codecs can not be added to the media profiles list.

The following table defines the ITU-T to IANA codec mappings.

ITU-T	IANA
g711Ulaw64k	PCMU
g711Alaw64k	PCMA
g726	G726
G7231	G723
g728	G728
g729wAnnexB	G729
g729	G729 fmtp:18 annexb=no
H261VideoCapability	H261
H263VideoCapability	H263
t38Fax	T38

SDP Anonymization

In order to provide an added measure of security, the Oracle Enterprise Session Border Controller's topology-hiding capabilities include SDP anonymization. Enabling this feature gives the Oracle Enterprise Session Border Controller the ability to change or modify certain values in the SDP so that malicious parties will be unable to learn information about your network topology.

To do this, the Oracle Enterprise Session Border Controller hides the product-specific information that can appear in SDP o= lines and s= lines. This information can include usernames, session names, and version fields. To resolve this issues, the Oracle Enterprise Session Border Controller makes the following changes when you enable SDP anonymization:

- Sets the session name (or the s= line in the SDP) to s=-
- Sets the username in the origin field to -SBC
- Sets the session ID in the origin field to an integer of incrementing value

Note that for mid-call media changes, the session identifier is not incremented.

To enable this feature, you set a parameter in the media manager configuration.

SDP Anonymization Configuration

To enable SDP anonymization:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type media-manager again to access the media manager configuration, and press Enter.

   ```
   ACMEPACKET(media-manager)# media-manager
   ACMEPACKET(media-manager-config)#
   ```

4. anonymous-sdp—Set this parameter to enabled to use the SDP anonymization feature. When you leave this parameter empty the feature is turned off. The default value is disabled. The valid values are:

   - enabled | disabled

5. Save and activate your configuration.

Unique SDP Session ID

Codec negotiation can be enabled by updating the SDP session ID and version number. The media-manager option, unique-sdp-id enables this feature.

With this option enabled, the Oracle Enterprise Session Border Controller will hash the session ID and IP address of the incoming SDP with the current date/time of the Oracle Enterprise Session Border Controller in order to generate a unique session ID.

Unique SDP Session ID Configuration

To enable unique SDP session ID in media-manager:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. options—Set the options parameter by typing options, a Space, the option name unique-sdp-id with a plus sign in front of it, and then press Enter.

```
ACMEPACKET(media-manager)# options +unique-sdp-id
```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

4. Save and activate your configuration.

TCP Synchronize Attack Prevention

This section explains how the Oracle Enterprise Session Border Controller protects itself from a Transmission Control Protocol (TCP) synchronize (SYN) packet flooding attack sourced from a remote hostile entity.

SIP and H.323 signaling can be configured on the Oracle Enterprise Session Border Controller to be TCP protocol-based. In this configuration, the Oracle Enterprise Session Border Controller can be a target of a TCP SYN attack. The Oracle Enterprise Session Border Controller C is able to service new call requests throughout the duration of an attack

About SYN

SYN is used by TCP when initiating a new connection to synchronize the sequence numbers on two connecting computers. The SYN is acknowledged by a SYN-ACK by the responding computer. After the SYN-ACK, the client finishes establishing the connection by responding with an ACK message. The connection between the client and the server is then open, and the service-specific data can be exchanged between the client and the server.

A SYN flood is a series of SYN packets from forged IP addresses. The IP addresses are chosen randomly and do not provide any hint of the attacker's location. The SYN flood keeps the server's SYN queue full. Normally this would force the server to drop connections. A server that uses SYN cookies, however, will continue operating normally. The biggest effect of the SYN flood is to disable large windows.

Server Vulnerability

Vulnerability to attack occurs when the server has sent a SYN-ACK back to client, but has not yet received the ACK message; which is considered a half-open connection. The server has a data structure describing all pending connections built in its system memory. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections.

The attacking system sends SYN messages to the server that appear to be legitimate, but in fact reference a client that is unable to respond to the SYN-ACK messages. The final ACK message is never sent to the server.

The half-open connections data structure on the server fills and no new incoming connections are accepted until the table is emptied out. Typically there is a timeout associated with a pending connection (the half-open connections will eventually expire and the server will recover). But the attacking system can continue sending IP-spoofed packets requesting new connections faster than the server can expire the pending connections. The server has difficulty in accepting any new incoming network connections.

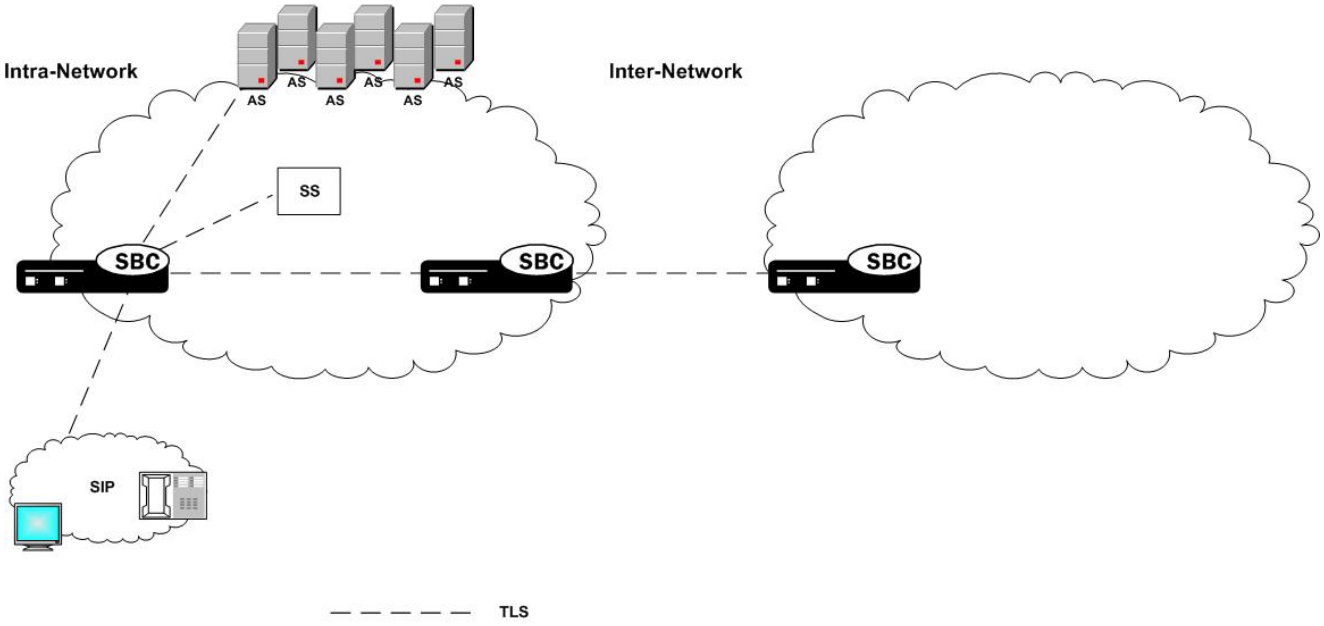
Configuring TCP SYN Attack Prevention

No configuration is necessary to enable TCP SYN attack prevention. Internal TCP protocol changes were made to provide protection.

Transport Layer Security

The Oracle Enterprise Session Border Controller provides support for Transport Layer Security (TLS) for SIP, which can be used to protect user and network privacy by providing authentication and guaranteeing the integrity for communications between the Oracle Enterprise Session Border Controller and the following:

- Another device in your network infrastructure (intra-network)
- Another Oracle Enterprise Session Border Controller when you are using a peering application (inter-network) for interior network signaling security
- An endpoint for authentication before allowing SIP messaging to take place
-



The Oracle Enterprise Session Border Controller and TLS

The Oracle Enterprise Session Border Controller ’s TLS functionality depends on the presence of the Security Service Module (SSM) for hardware acceleration of encryption and decryption and random media generation. The SSM is a plug-on module that can be added to your Oracle Enterprise Session Border Controller chassis given the installation of the necessary bootloader and minimum hardware revision levels.

With the requisite hardware revision levels, the plug-on unit can be added to your Oracle Enterprise Session Border Controller in the field by qualified personnel. This provision makes upgrades fast, forgoing the need for you to return your Oracle Enterprise Session Border Controller to Oracle manufacturing for hardware upgrade. When your Oracle Enterprise Session Border Controller is upgraded with the SSM card that supports TLS, a new CLEI code will be added to your chassis; the code will also appear on the SSM card (also referred to as the plug-on unit) and visible if

the system's chassis cover is opened. New Oracle Enterprise Session Border Controller s outfitted with the SSM card will have the code labels already affixed in all required locations.

TLS support will not behave in the manner described here if you do not have the SSM component installed on your Oracle Enterprise Session Border Controller , because it is the presence of this hardware that enables the TLS software support.

The accelerator card performs:

- RSA
- Diffie-Hellman
- DES
- 3DES
- 40/128 bit ARCFOUR
- AES256
- Random number generation

TLS Features

The Oracle Enterprise Session Border Controller supports the following TLS features:

- TLSv1/SSLv3
- RFC 3261 specific SIPS and TLS support in SIP
- Importing X509v3 certificates in PKCS-7/X509v3 PEM/Base64 format
- Generating a private key and a certificate request in PKCS-10 PEM/Base64 format
- Displaying imported certificates in text format
- Configuration verification, including verification that all dependencies are resolved
- Connection reuse draft (draft-ietf-sip-connect-reuse-03.txt)
- S-CX6.3.0 allows a maximum of 60000 TLS and 60000 TCP connections where each TLS connection also consumes one TCP connection.
- As of S-CX6.3.0M3, these capacities are raised to 100000 TLS connections and 120000 TCP connections.
- HA for TLS—When the active system in an HA node fails, the standby has the same TLS-related configuration, which is accomplished through configuration checkpointing as described in the HA Nodes chapter.

  Existing active calls are not affected by a failover—Enduser experiences no interruption or disturbance in service. SIP signaling messages sent over the connection following failover do not impact the active call.

  New calls, new TLS connections are made

The Oracle Enterprise Session Border Controller does not support certificate revocation listing handling.

Domestic and International Versions

There are two versions of the Acme Packet OS that support TLS: a U.S. version and an international version. Two versions exist because of the laws governing the strength of algorithms that can be shipped domestically and internationally. If you require further information, consult with your ACMEPACKET sales representative directly.

Supported Encryption

The Oracle Enterprise Session Border Controller provides support for TLSv1 and SSLv3 encryption.

Note: We do not support RC4 ciphers on the Net-Net 3800 or the Net-Net 4500. We do continue to support RC4 ciphers on the Net-Net 4250 for backwards-compatibility purposes.

TLSv1 Ciphers

The Oracle Enterprise Session Border Controller supports the TLS v1 cipher suites listed in this section.

For encryption, the Oracle Enterprise Session Border Controller supports: AES-128, AES-256, 3DES, DES and ARC4 (40 and 128 bit) algorithms. It also supports:

- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_RSA_WITH_DES_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- ALL [default]
- NONE

Mapping SSL3 to TLSv1 Ciphers

The following table shows the mapping of SSL3 ciphers to TLSv1 ciphers:

SSL3	TLSv1
SSL_RSA_WITH_NULL_MD5	TLS_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA	TLS_RSA_WITH_NULL_SHA
SSL_RSA_WITH_RC4_128_MD5	TLS_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA	TLS_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA	TLS_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	TLS_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

Note: The Oracle Enterprise Session Border Controller supports TLS_RSA_WITH_NULL_MD5 and TLS_RSA_WITH_NULL_SHA although neither does any encryption. These ciphers might be used for debugging purposes, and should not be deployed.

Signaling Support

The Oracle Enterprise Session Border Controller ’s TLS functionality supports SIP and SIPS. In addition, the Oracle Enterprise Session Border Controller can accommodate a mixture of TLS and non-TLS sessions within a realm as because a request for TLS is controlled by the endpoint (TLS UA).

DoS Protection

The Oracle Enterprise Session Border Controller provides the following forms of DoS protection from:

- Too many simultaneous TLS connections being requested by a single IP address.

  The Oracle Enterprise Session Border Controller limits the number of TLS connections from a single IP address; you can set a maximum simultaneous number of TCP/TLS connections a SIP interface will allow from a single IP address.
- Too many simultaneous TLS connections being requested by limiting the maximum number of connections for a SIP interface.

In other words, the maximum simultaneous TCP/TLS connections a SIP interface will allow in aggregate from all IP addresses served by that signaling interface.

- Endpoints establishing TCP/TLS connections that never send any messages (application layer messages; once the TLS handshake completes).

  This protection is triggered by inactivity, measured by lack of any message from a peer. The value specified for this timer is in seconds.
- Endpoints requesting an initial registration that never send messages thereafter.

  Note: It is expected that whenever an endpoint establishes a TCP/TLS connection, it will keep the connection active by sending additional messages or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level "ERROR" is generated.
- Malformed packets by counting and limiting the maximum number of malformed packets.

  Whenever the Oracle Enterprise Session Border Controller receives an invalid TLS message, it increments the internal invalid signalling threshold counter. When that counter reaches the configured value, the Oracle Enterprise Session Border Controller denies the endpoints for the configured deny period. This also requires configuration of tolerance window in media manager.

Endpoint Authentication

The Oracle Enterprise Session Border Controller does not operate as a CA. Instead, the Oracle Enterprise Session Border Controller ’s TLS implementation assumes that you are using one of the standard CAs for generating certificates:

- Verisign
- Entrust
- Thawte
- free Linux-based CA (for example, openssl)

Note: Self-signed certificates are available only as an option for MSRP connections

The Oracle Enterprise Session Border Controller can generate a certificate request in PKCS10 format and to export it. It can also import CA certificates and a Oracle Enterprise Session Border Controller certificate in the PKCS7/X509 PEM format.

The Oracle Enterprise Session Border Controller generates the key pair for the certificate request internally. The private key is stored as a part of the configuration in 3DES encrypted form (with an internal generated password) and the public key is returned to the user along with other information as a part of PKCS10 certificate request.

The Oracle Enterprise Session Border Controller supports the option of importing CA certificates and marking them as trusted. However, the Oracle Enterprise Session Border Controller only authenticates client certificates that are issued by the CAs belonging to its trusted list. If you install only a specific vendor's CA certificate on the Oracle Enterprise Session Border Controller , it authenticates that vendor's endpoints. Whether the certificate is an individual device certificate or a site-to-site certificate does not matter because the Oracle Enterprise Session Border Controller authenticates the signature/public key of the certificate.

Key Usage Control

You can configure the role of a certificate by setting key usage extensions and extended key usage extensions. Both of these are configured in the certificate record configuration.

Key Usage List

This section defines the values you can use (as a list) in the **key-usage-list** parameter. You can configure the parameter with more than one of the possible values.

Value	Description
digitalSignature (default with keyEncipherment)	Used when the subject public key is used with a digital signature mechanism to support security services other than non-repudiation, certificate signing, or revocation information signing. Digital signature mechanisms are often used for entity authentication and data origin authentication with integrity.
nonRepudiation	Used when the subject public key is used to verify digital signatures that provide a non-repudiation service protecting against the signing entity falsely denying some action, excluding certificate or CRL signing.
keyEncipherment (default with digitalSignature)	Used with the subject public key is used for key transport. (For example, when an RSA key is to be used for key management.)
dataEncipherment	Used with the subject public key is used for enciphering user data other than cryptographic keys.
keyAgreement	Used with the subject public key is used key agreement. (For example, when a Diffie-Hellman key is to be used for a management key.)
encipherOnly	The keyAgreement type must also be set. Used with the subject public key is used only for enciphering data while performing key agreement.
decipherOnly	The keyAgreement type must also be set. Used with the subject public key is used only for deciphering data while performing key agreement.

Extended Key Usage List

This section defines the values you can use in the extended-key-usage-list parameter.

Value	Description
serverAuth (default)	Used while the certificate is used for TLS server authentication. In Oracle Enterprise Session Border Controller access-side deployments, the system typically acts as a TLS server accepting TLS connections. You might use this setting while generating the end-entity-cert.
clientAuth	Used while the certificate is used for TLS client authentication. In Oracle Enterprise Session Border Controller core-side deployments, the system typically acts as a TLS client initiating TLS connections. You might use this setting while generating the end-entity-cert.

Configuring TLS

This section explains how to configure your Oracle Enterprise Session Border Controller for TLS support.

Process Overview

In summary, you need to take the following steps to enable your Oracle Enterprise Session Border Controller for TLS.

1. Make sure that your Oracle Enterprise Session Border Controller has the appropriate hardware installed and that you have obtained an enabled the licenses related to TLS support. (Note that the Acme Packet 4250 does not require an additional license for TLS support.)
2. Configure certificates.

3. Configure the specific parameters related to TLS.

Configuring Certificates

Configuring certificates is a three-step process:

1. Create a certificate record configuration on the Oracle Enterprise Session Border Controller
2. Generate a certificate request by the Oracle Enterprise Session Border Controller and save the configuration
3. Import the certificate record into the Oracle Enterprise Session Border Controller and save the configuration

Configuring the Certificate Record

The certificate record configuration represents either the end-entity or the CA certificate on the Oracle Enterprise Session Border Controller. If it is used to present an end-entity certificate, a private key should be associated with this certificate record configuration using the ACLI generate-certificate-request command. A certificate, provided by a CA in response to a certificate request, can be imported to a certificate record configuration using the ACLI import-certificate command.

No private key should be associated with the certificate record configuration if it was issued to hold a CA certificate.

Note: There is no need to create a certificate record when importing a CA certificate or certificate in pkcs12 format.

The following is sample of the certificate record configuration parameters as seen in the ACLI.

```
certificate-record
name             certificate record name
country          country name
state            state name
locality         locality name
organization     organization name
unit             organization unit
common-name      common name
key-size         key size
alternate-name   alternate name
trusted          certificate-record trusted or not
```

To enter a certificate record using the ACLI configuration menu:

1. Access the **certificate-record** configuration element.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# certificate record
   ACMEPACKET(certificate-record)#
   ```

2. name—Provide the required name of this certificate record object.
3. **country**, **state**, **locality**, **organization**, **unit**, and **common-name**—Use these parameters to identify the certificate's subject.

   ```
   ACMEPACKET(certificate-record)# country us
   ACMEPACKET(certificate-record)# state ca
   ACMEPACKET(certificate-record)# locality "San Francisco"
   ACMEPACKET(certificate-record)# organization "Office of the CTO"
   ACMEPACKET(certificate-record)# unit cyzygy.com
   ACMEPACKET(certificate-record)# common-name www.cyzygy.org/
   emailAddress=cto@cyzygy.org
   ```

   Based on this ACLI sequence, the subject field of the resulting CA-issued certificate reads as follows.

   ```
   Subject: C=US, ST=California, L=San Francisco, O=Office of the CTO
   OU=Cyzygy,
   CN=www.cyzygy.org/emailAddress=cto@cyzygy.org
   ```

4. key-size—Enter the size of the key for the certificate. The default value is 1024. The valid range is:

   - 512 | 1024 (default) | 2048

5. alternate-name—Optionally provide one or more alternative names for the certificate holder. The Subject Alternative Name certificate extension is defined in section 4.2.1.6 of RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. This extension allows one or more identities to be bound to the subject of a certificate. As a result, each subject alternative name is considered as an alias for the certificate subject. These aliases may be included in addition to, or in place of the identity in the subject field of the certificate, meaning that a certificate can contain a subject name and one or more aliases, or no subject name and one or more aliases. A certificate cannot lack both a subject name and an alias. As specifically defined in RFC 5280, subject alternative names can take the form of an e-mail address, an IP address (IPv4 or IPv6), a DNS address, a Registered ID (RID), a Distinguished Name (DN), or a URI.

   RFC also 5280 allows future support for other alternative name forms, but only if such forms are specified in an IETF RFC.

   Most alternative names take one of the following three formats: IP address (which requires an IP: prefix), DNS (which requires a DNS: prefix), or email (which requires an email: prefix).

   Example:

```
CMEPACKET(certificate-record)# alternate-name IP:192.168.12.101
ACMEPACKET(certificate-record)#

ACMEPACKET(certificate-record)# alternate-name IP:13::17
ACMEPACKET(certificate-record)#

ACMEPACKET(certificate-record)# alternate-name DNS:sg1a.ba.de
ACMEPACKET(certificate-record)#

ACMEPACKET(certificate-record)# alternate-name email:my@other.address
ACMEPACKET(certificate-record)#

ACMEPACKET(certificate-record)# alternate-name copy,email:my@other.address
ACMEPACKET(certificate-record)#
```

   The email form can include a special copy value that includes any email addresses contained in the certificate subject name in the extension.

   For other formats, consult RFC 5280, later RFCs that define future formats, or the OpenSSL documentation available at: http://www.openssl.org/docs/apps/x509v3_config.html

6. trusted—Leave this parameters set to enabled to make the certificate trusted. Enter disabled to make this certificate untrusted. The default value is enabled. The valid values are:

   - enabled | disabled

7. key-usage-list—Enter the usage extensions you want to use with this certificate record. This parameter can be configured with multiple values, and it defaults to the combination of digitalSignature and keyEncipherment. For a list of possible values and their descriptions, see the section “Key Usage Control” in the *Oracle Communications Session Border Controller Configuration Guide*.
8. extended-key-usage-list—Enter the extended key usage extensions you want to use with this certificate record. The default is serverAuth. For a list of possible values and their descriptions, see the section “Key Usage Control” in the *Oracle Communications Session Border Controller Configuration Guide*.
9. Type done and press Enter.

```
ACMEPACKET(certificate-record)# done
```

10. Type **done** to save your configuration.

You create TLS profiles, using your certificate records, to further define the encryption behavior and create the configuration element that you can then apply to a SIP interface.

Generating a Certificate Request

Using the ACLI **generate-certificate-request** command allows you to generate a private key and a certificate request in PKCS10 PEM format. You take this step once you have configured a certificate record.

The Oracle Enterprise Session Border Controller stores the private key that is generated in the certificate record configuration in 3DES encrypted form with in internally generated password. The PKCS10 request is displayed on the screen in PEM (Base64) form.

You use this command for certificate record configurations that hold end-entity certificates. If you have configured the certificate record to hold a CA certificate, then you do not need to generate a certificate request because the CA publishes its certificate in the public domain. You import a CA certificate by using the ACLI **import-certificate** command.

This command sends information to the CA to generate the certificate, but you cannot have Internet connectivity from the Oracle Enterprise Session Border Controller to the Internet. You can access the internet through a browser such as Internet Explorer if it is available, or you can save the certificate request to a disk and then submit it to the CA.

To run the applicable command, you must use the value you entered in the name parameter of the certificate record configuration. You run the command from main Superuser mode command line:

```
ACMEPACKET# generate-certificate-request acmepacket
Generating Certificate Signing Request. This can take several minutes...
-----BEGIN CERTIFICATE REQUEST-----
MIIDHzCCAoigAwIBAgIIAhMCUACEAHEwDQYJKoZIhvcNAQEFBQAwcDELMAkGA1UE
BhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMQ4w
DAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lwaXQgVGVzdCBDZXJ0aWZpY2F0ZSBB
dXRob3JpdHkwHhcNMDUwNDEzMjEzNzQzWhcNMDgwNDEyMjEzNzQzWjBUMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCTUExEzARBgNVBAcTCkJ1cmxpbmd0b24xFDASBgNV
BAoTC0VuZ2luZWVyaW5nMQ0wCwYDVQQDEwRhY21lMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCXjIeOyFKAUB3rKkKK/+59LT+rlGuW7Lgc1V6+hfTSr0co+ZsQ
bHFUWAA15qXUUBTLJG13QN5VfG96f7gGAbWayfOS9Uymold3JPCUDoGgb2E7m8iu
vtq7gwjSeKNXAw/y7yWy/c04FmUD2U0pZX0CNIR3Mns5OAxQmq0bNYDhawIDAQAB
o4HdMIHaMBEGA1UdEQQKMAiCBnBrdW1hcjAJBgNVHRMEAjAAMB0GA1UdDgQWBBTG
tpodxa6Kmmn04L3Kg62t8BZJHTCBmgYDVR0jBIGSMIGPgBRrRhcU6pR2JYBUbhNU
2qHjVBShtqF0pHIwcDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEx
ETAPBgNVBAcTCFNhbiBKb3NlMQ4wDAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lw
aXQgVGVzdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHmCAQAwDQYJKoZIhvcNAQEFBQAD
gYEAbEs8nUCi+cA2hC/lM49Sitvh8QmpL81KONApsoC4Em24L+DZwz3uInoWjbjJ
QhefcUfteNYkbuMH7LAK0hnDPvW+St4rQGVK6LJhZj7/yeLXmYWIPUY3Ux4OGVrd
2UgV/B2SOqH9Nf+FQ+mNZOlL7EuF4IxSz9/69LuYlXqKsG4=
-----END CERTIFICATE REQUEST-----;
WARNING: Configuration changed, run save-config command.
ACMEPACKET# save-config
Save-config received, processing.
waiting 1200 for request to finish
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot-activate'
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting 12000 for request to finish
Add LI flows
LiSysClientMgr::handleNotifyReq
H323 Active Stack Cnt: 0
Request to 'ACTIVATE-CONFIG' has finished
Activate Complete
ACMEPACKET#
```

Importing a Certificate Using the ACLI

For an end-entity certificate, once a certificate is generated using the ACLI generate-certificate-request command, that request should be submitted to a CA for generation of a certificate in PKCS7 or X509v3 format. When the certificate has been generated, it can be imported into the Oracle Enterprise Session Border Controller using the import-certificate command.

The syntax is:

```
ACMEPACKET # import-certificate [try-all|pkcs7|x509] [certificate-record file-
name]
```

To import a certificate:

1. When you use the import-certificate command, you can specify whether you want to use PKCS7 or X509v3 format, or try all. In the command line, you enter the command, the format specification, and the name of the certificate record.

   ```
   ACMEPACKET# import-certificate try-all acme
   ```

   The following will appear:

   ```
   Please enter the certificate in the PEM format.
   Terminate the certificate with ";" to exit.......
   -----BEGIN CERTIFICATE-----
   MIIDHzCCAoigAwIBAgIIAhMCUACEAHEwDQYJKoZIhvcNAQEFBQAwcDELMAkGA1UE
   BhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMQ4w
   DAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lwaXQgVGVzdCBDZXJ0aWZpY2F0ZSBB
   dXRob3JpdHkwHhcNMDUwNDEzMjEzNzQzWhcNMDgwNDEyMjEzNzQzWjBUMQswCQYD
   VQQGEwJVUzELMAkGA1UECBMCTUExEzARBgNVBAcTCkJ1cmxpbmd0b24xFDASBgNV
   BAoTC0VuZ2luZWVyaW5nMQ0wCwYDVQQDEwRhY21lMIGfMA0GCSqGSIb3DQEBAQUA
   A4GNADCBiQKBgQCXjIeOyFKAUB3rKkKK/+59LT+rlGuW7Lgc1V6+hfTSr0co+ZsQ
   bHFUWAA15qXUUBTLJG13QN5VfG96f7gGAbWayfOS9Uymold3JPCUDoGgb2E7m8iu
   vtq7gwjSeKNXAw/y7yWy/c04FmUD2U0pZX0CNIR3Mns5OAxQmq0bNYDhawIDAQAB
   o4HdMIHaMBEGA1UdEQQKMAiCBnBrdW1hcjAJBgNVHRMEAjAAMB0GA1UdDgQWBBTG
   tpodxa6Kmmn04L3Kg62t8BZJHTCBmgYDVR0jBIGSMIGPgBRrRhcU6pR2JYBUbhNU
   2qHjVBShtqF0pHIwcDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEx
   ETAPBgNVBAcTCFNhbiBKb3NlMQ4wDAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lw
   aXQgVGVzdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHmCAQAwDQYJKoZIhvcNAQEFBQAD
   gYEAbEs8nUCi+cA2hC/lM49Sitvh8QmpL81KONApsoC4Em24L+DZwz3uInoWjbjJ
   QhefcUfteNYkbuMH7LAK0hnDPvW+St4rQGVK6LJhZj7/yeLXmYWIPUY3Ux4OGVrd
   2UgV/B2SOqH9Nf+FQ+mNZOlL7EuF4IxSz9/69LuYlXqKsG4=
   -----END CERTIFICATE-----;
   Certificate imported successfully....
   WARNING: Configuration changed, run "save-config" command.
   ```

2. Save your configuration.

   ```
   ACMEPACKET# save-config
   Save-Config received, processing.
   waiting 1200 for request to finish
   Request to 'SAVE-CONFIG' has Finished,
   Save complete
   Currently active and saved configurations do not match!
   To sync & activate, run 'activate-config' or 'reboot activate'.
   ```

3. Synchronize and activate your configurations.

   ```
   ACMEPACKET# activate-config
   Activate-Config received, processing.
   waiting 120000 for request to finish
   Add LI Flows
   LiSysClientMgr::handleNotifyReq
   H323 Active Stack Cnt:  0
   Request to 'ACTIVATE-CONFIG' has Finished,
   Activate Complete
   ACMEPACKET#
   ```

Importing a Certificate Using FTP

You can also put the certificate file in the directory /ramdrv and then executing the import-certificate command or by pasting the certificate in the PEM/Base64 format into the ACLI. If you paste the certificate, you might have to copy and paste it a portion at a time rather than pasting in the whole thing at once.

To import the certificate using FTP:

1. FTP the certificate file on to the Oracle Enterprise Session Border Controller (directory /ramdrv), let us say the name of the certificate file is cert.pem.

2. Once the certificate is successfully transferred to the Oracle Enterprise Session Border Controller , run the import-certificate command.

The syntax is:

```
ACMEPACKET# import-certificate [try-all|pkcs7|x509] [certificate-record file-
name]
```

Using the command will look like this when you have used FTP.

```
ACMEPACKET# import-certificate try-all acme cert.pem
Certificate imported successfully....
WARNING: Configuration changed, run "save-config" command.
```

1. Save your configuration.

```
ACMEPACKET# save-config
Save-Config received, processing.
waiting 1200 for request to finish
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.
```

2. Synchronize and activate your configurations.

```
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting 120000 for request to finish
Add LI Flows
LiSysClientMgr::handleNotifyReq
H323 Active Stack Cnt:  0
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ACMEPACKET#
```

Configuring a TLS Profile

The TLS profile configuration has been added to the security section of the ACLI's configure terminal menu. This configuration holds the information required to run SIP over TLS.

In the ALCI menu for this configuration, the parameters appear as follows:

```
tls-profile
name                      tls profile name
end-entity-certificate    end entity certificate for the TLS connection
trusted-ca-certificates   list of trusted certificate records
cipher-list           list of ciphers
verify-depth          maximum length of the certificate chain
mutual-authenticate   mutually authenticate
```

To configure a TLS profile:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type security and press Enter to access the session-router path.

```
ACMEPACKET(configure)# security
```

3. Type tls-profile and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)#
```

4. name—Enter the name of the TLS profile. This parameter is required; you cannot leave it empty.

5. end-entity-certificate—Enter the name of the entity certification record.
6. trusted-ca-certificates—Enter the names of the trusted CA certificate records.
7. cipher-list—Either use the default ALL, or enter a list of ciphers you want to support.
8. verify-depth—Specify the maximum depth of the certificate chain that will be verified. The default value is 10. The valid range is:
   - Minimum—0
   - Maximum—10
9. mutual-authenticate—Define whether or not you want the Oracle Enterprise Session Border Controller to mutually authenticate the client. The default value is disabled. The valid values are:
   - enabled | disabled
10. tls-version—Enter the TLS version you want to use with this TLS profile. Valid values are TLSv1, SSLv3, and compatibility (default).
11. Save your work.
12. Exit out to the configuration terminal menu to apply the TLS profile.

```
ACMEPACKET(tls-profile)# exit
ACMEPACKET(security)# exit
ACMEPACKET(configure)#
```

Applying a TLS Profile

To apply the TLS profile, you need to specify it for the SIP interface with which it will be used. You must take this step from within the SIP interface configuration.

1. Type session-router and press Enter to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

2. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

3. Select the existing SIP interface to which you want to apply the TLS profile. If you do not know the same of the profile, press Enter again after you use the select command to see a list of all SIP interfaces. Type in the number corresponding to the SIP interface you want to select, and press Enter. You will then be modifying that SIP interface.

```
ACMEPACKET(sip-interface)# select
```

4. Type sip-ports and Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-interface)# sip-ports
ACMEPACKET(sip-port)#
```

5. transport-protocol—Change the transport protocol to TLS.

```
ACMEPACKET(sip-interface)# transport-protocol tls
```

6. tls-profile—Enter the name of the TLS profile you want applied. This is the same value you enter for the name parameter in the TLS profile configuration. This profile will be applied when the transport protocol is TLS.

```
ACMEPACKET(sip-interface)# tls-profile acmepacket
```

7. Save your updated SIP interface configuration.

Reusing a TLS Connection

TheOracle Enterprise Session Border Controller supports TLS connection reuse if and when an alias is included in the Via header by the originator of the TLS connection. When this is the case, the Oracle Enterprise Session Border Controller reuses the same connection for any outgoing request from the Oracle Enterprise Session Border Controller .

Keeping Pinholes Open at the Endpoint

The Oracle Enterprise Session Border Controller provides configurable TCP NAT interval on a per-realm basis. You need to configure a NAT interval for the applicable realm to support either all conforming or all non-conforming endpoints.

- Conforming endpoints use the draft-jennings sipping-outbound-01. It describes how to keep the endpoint keeps the connection alive.

  Note: Currently the endpoint uses REGISTER.

- Non-conforming endpoints have short NAT interval, where the HNT application with the TCP connection for TLS operates as it does for regular TCP. We give the UA a shorter expires time so that it refreshes frequently, implicitly forcing the UA to keep the TVP socket open and reuse it for further requests (in-dialog or out-of-dialog). Regular requests using TLS sent from the Oracle Enterprise Session Border Controller to the UA reuse the same TCP connection so that further TLS certificate exchanges are not required.

Viewing Certificates

You can view either a brief version or detailed information about the certificates.

Brief Version

Obtaining the brief version uses this syntax, and will appear like the following example:

```
ACMEPACKET# show security certificates brief acmepacket
certificate-record:acmepacket
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            02:13:02:50:00:84:00:71
        Issuer:
            C=US
            ST=California
            L=San Jose
            O=sipit
            OU=Sipit Test Certificate Authority
        Subject:
            C=US
            ST=MA
            L=Burlington
            O=Engineering
            CN=acme
ACMEPACKET#
```

Detailed Version

Obtaining the detailed version uses this syntax, and will appear like the following example:

```
ACMEPACKET# show security certificates detail acmepacket
certificate-record:acmepacket
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            02:13:02:50:00:84:00:71
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:
            C=US
            ST=California
            L=San Jose
            O=sipit
            OU=Sipit Test Certificate Authority
```

```
        Validity
            Not Before: Apr 13 21:37:43 2005 GMT
            Not After : Apr 12 21:37:43 2008 GMT
        Subject:
            C=US
            ST=MA
            L=Burlington
            O=Engineering
            CN=acme
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:pkumar
            X509v3 Basic Constraints:
                CA:FALSE
ACMEPACKET#
```

Host Certificate Retrieval via SNMP

When a security certificate is installed locally on the Oracle Enterprise Session Border Controller , you can poll the expiration of the certificate using the apSecurityCertificateTable.

You can configure the Oracle Enterprise Session Border Controllerto generate the apSecurityCertExpiredNotification trap once a certificate has expired. The number of minutes between notifications sent is configured in the security-config parameter local-cert-trap-int.

To send a warning of expiration, you can set the security-config parameter local-cert-exp-warn-period to the number of days before the locally installed certificate expires in which you would like a warning.

Host Certificate Retrieval Configuration

To configure the Oracle Enterprise Session Border Controller to generate traps when a certificate has or is about to expire:

1. Navigate to the security-config configuration element.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security#
   ACMEPACKET(security)# security-config
   ACMEPACKET(security-config)#
   ```

2. Set the local-cert-exp-warn-period parameter to the number of days before the locally installed certificate expires in order to receive a warning. A value of 0 disables the trap.

   ```
   ACMEPACKET(security-config)# local-cert-exp-warn-period 3
   ACMEPACKET(security-config)#
   ```

3. Set the local-cert-trap-int parameter for the number of minutes between notifications sent once a certificate has expired. A value of 0 disables the warning trap.

   ```
   ACMEPACKET(security-config)# local-cert-exp-trap-int 15
   ACMEPACKET(security-config)#
   ```

4. Use done, exit, and verify-config to complete required configuration.

Denial of Service for TLS

This section explains the DoS for TLS feature. With this feature, the Oracle Enterprise Session Border Controller can provide protection from TCP/TLS message flood by limiting the number of connections from an end point and by limiting the number of simultaneous TCP/TLS connections to a SIP interface.

The Oracle Enterprise Session Border Controller protects against a flood of invalid TLS messages and against end points establishing TCP/TLS connections or doing an initial registration without then sending any messages. The Oracle Enterprise Session Border Controller protects against:

- Too many simultaneous TLS connections being requested by a single IP address by limiting the number of TLS connections from a single IP address. There is a maximum simultaneous number of TCP/TLS connections a SIP interface will allow from a single IP address.
- Too many simultaneous TLS connections being requested by limiting the maximum number of connections for a SIP interface. There is a maximum number of simultaneous TCP/TLS connections a SIP interface will allow in aggregate from all IP addresses served by that signaling interface.
- End points establishing TCP/TLS connections without then sending any messages (application layer messages post TLS handshake complete). Triggered by inactivity as measured by lack of any message from this peer.
- End points doing an initial registration without then sending any messages.

  This timer could be used by the administrator to detect errors with the SIP configuration. It is expected that whenever an end point establishes a TCP/TLS connection, the end point will keep the connection active by sending messages with REGISTER or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level ERROR is generated.)
- Malformed packets by counting and limiting the maximum number of malformed packets. Whenever an invalid TLS message is received, the internal counter corresponding to invalid-signal-threshold is incremented. When the invalid signal threshold reaches the configured value, the end point will be denied for the configured deny period. (Also requires configuration of the tolerance window in media manager.)
- The max-incoming-conns parameter is well under the maximum number of TLS connections supported by the system. You can set this parameter to it's maximum value of 20000. If you need more than 20000 TLS connections available on this SIP interface, you must set max-incoming-conns to 0 which allows up to the system maximum number of TLS connections, taken on a first come first served basis, on this SIP interface.

DoS for TLS Configuration

You configure the SIP interface and the realm to support DoS for TLS.

DoS protection for TLS Connections on the SIP Interface Configuration

To configure the DoS protection for TCP/TLS connections on a SIP interface:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-interface and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.

4. max-incoming-conns—Enter the maximum number of simultaneous TCP/TLS connections for this SIP interface. The default value is zero (0) which disables any limit to the number of simultaneous TCP/TLS connections on this SIP interface. The valid range is:

   - Minimum—0
   - Maximum—20000

5. per-src-ip-max-incoming-conns—Enter the maximum number of connections allowed from an end point.The default value is zero (0). The default disables the parameter. The valid range is:

   - Minimum—0
   - Maximum—20000

   Note: To make this parameter effective, you need to set the realm's access-control-trust-level to low or medium.

6. inactive-conn-timeout—Enter the time in seconds you want a connection from an endpoint discontinued. This provides protection from end points doing an initial registration without sending any messages. The default value is zero (0). The default disables the parameter. The valid range is:

   - Minimum—0
   - Maximum—999999999

7. Save and activate your configuration.

Configuring the SIP Configuration

To configure the SIP configuration:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type sip-config and press Enter. The system prompt changes.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

   From this point, you can configure SIP configuration parameters. To view all sip-config parameters, enter a ? at the system prompt.

4. inactive-dynamic-conn—Enter the time in seconds after which the Oracle Enterprise Session Border Controller tears down inactive dynamic TCP connections. Inactive is defined as not transporting any traffic. This protects against endpoints establishing TCP/TLS connections and then not sending messages. The default value is 32. The valid range is:

   - Minimum—0
   - Maximum—999999999

   Note: Setting this parameter to 0 disables this parameter.

   Because the Oracle Enterprise Session Border Controller first establishes a TCP connection, then the TLS connection it waits twice the value entered here after the initiation of a TLS connection before tearing down the connection.

   After an endpoint establishes a TCP/TLS connection, it is supposed to keep the connection active by sending messages or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level ERROR is generated.

Configuring the Realm

To configure the realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the media-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

   From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

4. deny-period—Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is 30. The valid range is:

- Minimum—0
- Maximum—4294967295

5. invalid-signal-threshold— Enter the number of invalid TLS signaling messages that trigger host demotion. The value you enter here is only valid when the trust level is low or medium. Available values are:

   - Minimum—Zero (0) is disabled.
   - Maximum—999999999

   If the number of invalid messages exceeds this value based on the tolerance window parameter, configured in the media manager, the host is demoted.

   The tolerance window default is 30 seconds. Bear in mind, however, that the system uses the same calculation it uses for specifying "recent" statistics in show commands to determine when the number of signaling messages exceeds this threshold. This calculation specifies a consistent start time for each time period to compensate for the fact that the event time, such as a user running a show command, almost never falls on a time-period's border. This provides more consistent periods of time for measuring event counts.

   The result is that this invalid signal count increments for two tolerance windows, 60 seconds by default, within which the system monitors whether or not to demote the host. The signal count for the current tolerance window is always added to the signal count of the previous tolerance window and compared against your setting.

6. access-control-trust-level—Set the trust level for the host within the realm. The default value is none. The valid values are:

   - none—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
   - low—Host can be promoted to the trusted list or demoted to the deny list.
   - medium—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
   - high—Host is always trusted.

7. Save and activate your configuration.

TLS Session Caching

Transport Layer Security (TLS) session caching allows the Oracle Enterprise Session Border Controller to cache key information for TLS connections, and to set the length of time that the information is cached.

When TLS session caching is not enabled, the Oracle Enterprise Session Border Controller and a TLS client perform the handshake portion of the authentication sequence in which they exchange a shared secret and encryption keys are generated. One result of the successful handshake is the creation of a unique session identifier. When an established TLS connection is torn down and the client wants to reinstate it, this entire process is repeated. Because the process is resource-intensive, you can enable TLS session caching to avoid repeating the handshake process for previously authenticated clients to preserve valuable Oracle Enterprise Session Border Controller resources.

When TLS session caching is enabled on the Oracle Enterprise Session Border Controller, a previously authenticated client can request re-connection using the unique session identifier from the previous session. The Oracle Enterprise Session Border Controller checks its cache, finds the session identifier, and reinstates the client. This process reduces the handshake to three messages, which preserves system resources.

If the client offers an invalid session identifier, for example, one that the Oracle Enterprise Session Border Controller has never seen or one that has been deleted from its cache, the system does not allow the re-connection. The system negotiates the connection as a new connection.

TLS Session Caching Configuration

TLS session caching is global for all TLS functions on your Oracle Enterprise Session Border Controller. A new global TLS configuration (tls-global) has been added to the system for this purpose.

To enable global TLS session caching:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type security and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type tls-global and press Enter.

```
ACMEPACKET(security)# tls-global
ACMEPACKET(tls-global)#
```

4. session-caching—Set the state for TLS session caching to enabled if you want to turn this feature on. The default value is disabled. The valid values are:

   - enabled | disabled

5. session-cache-timeout—Enter the time in hours that you want the Oracle Enterprise Session Border Controller to cache unique session identifiers so that previously authenticated clients can reconnect. The default value is 12. A value of 0 disables this parameter. The valid range is:

   - Minimum—0
   - Maximum—24

   If you set this parameter to 0, then cache entries will never age (and not be deleted from the cache unless you use the clear-cache tls command to delete all entries from the TLS cache). RFC 2246, *The TLS Protocol Version 1.0*, recommends that you set this parameter at the maximum, 24.

TLS Endpoint Certificate Data Caching

To provide a higher level of security for unified messaging (UM), the Oracle Enterprise Session Border Controller allows you configure enforcement profiles to cache data from TLS certificates. During the authentication process, the system caches the data so it can use that data in subsequent SIP message processing. Thus the Oracle Enterprise Session Border Controller can:

- Add custom SIP header populated with information from TLS certificates—When the Oracle Enterprise Session Border Controller receives an INVITE from a GW, it can write proprietary headers into the SIP message. It uses the certificate information the GW provided during the TLS authentication process with the Oracle Enterprise Session Border Controller to do so.
- Compare the host of the Request-URI with information from TLS certificates—When an INVITE is destined for the unified messaging server, the Oracle Enterprise Session Border Controller checks the domain of the Request-URI it has generated prior to HMR application. It does so to verify that the Request-URI matches the domain information the UM server provided during the TLS authentication process with the Oracle Enterprise Session Border Controller.

TLS endpoint certificate data caching can only applies to call-creating SIP INVITEs. The Oracle Enterprise Session Border Controller looks to the following configurations, in order, to apply an enforcement profile: session agent, realm, and SIP interface associated with the INVITE. As a final step, it checks the SIP profile for enforcement profile association.

Inserting Customized SIP Headers in an Outgoing INVITE

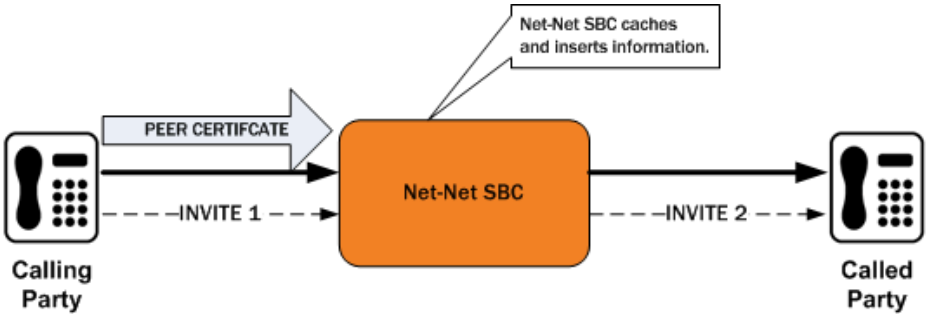
When the Oracle Enterprise Session Border Controller establishes a new TLS connection, it caches the following peer certificate attributes:

- Certificate Subject Name
- Certificate Subject Alternative Name (only DNS)

The Oracle Enterprise Session Border Controller constructs a customized P-Certificate-Subject-Common-Name SIP header and inserts the header into the outgoing INVITE with the Certificate Subject Name. The Oracle Enterprise Session Border Controller also constructs and inserts in the outgoing INVITE one or more P-Certificate-Subject-Alternative-Name SIP headers.

If you enable this capability and the incoming INVITE already has P-Certificate-Subject-Common-Name and P-Certificate-Subject-Alternative-Name headers, the Oracle Enterprise Session Border Controller strips them before inserting the new customized ones. It does so to avoid the risk of any attempt to spoof the headers and thereby gain unauthorized access to the UM server.

The following diagram shows a scenario where the calling party establishes a TLS connection with the Oracle Enterprise Session Border Controller. Because mutual authentication is enabled, the Oracle Enterprise Session Border Controller receives the peer certificate and caches required information from it. This information is inserted in the outgoing INVITE.



The peer certificate from the calling party during the TLS handshake with the Oracle Enterprise Session Border Controller looks like the following example.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9 (0x9)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=MA, L=Woburn, O=Smith Securities, OU=Certificate
Authority Dept, CN=Smith Certificate Authority/emailAddress=Smith@CA.com
        Validity
            Not Before: Dec 10 21:14:56 2009 GMT
            Not After : Jul 11 21:14:56 2019 GMT
        Subject: C=US, ST=MA, L=Burlington, O=Acme Packet, OU=Certificate
Authority Dept, CN=*.acme.com/emailAddress=ph1Client@acme.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            X509v3 Issuer Alternative Name:
            email:Smith@CA.com
            X509v3 Subject Alternative Name:
            DNS:gw1.acme.com, DNS:gw3.ano.com, DNS:gw2.some.com
            X509v3 Key Usage: critical
            Digital Signature, Key Encipherment
    Signature Algorithm: sha1WithRSAEncryption
```

The outgoing SIP INVITE (INVITE 2 in the diagram) looks like the following sample. Bold text shows where the Oracle Enterprise Session Border Controller uses information from the certificate.

```
INVITE sip:222222@acme.com:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.27.113:5060;branch=z9hG4bK4jmg29cmm8l0cg7smmrn85o4q7
From: 111111 <sip:111111@acme.com>;tag=_ph1_tag
To: 222222 <sip:222222@acme.com>
Call-ID: _1-2_call_id-10147@acme.com-1-
CSeq: 1 INVITE
Contact: <sip:111111@172.16.27.113:5060;transport=udp>
P-Certificate-Subject-Common-Name: *.acme.com
P-Certificate-Subject-Alternative-Name: gw1.acme.com
P-Certificate-Subject-Alternative-Name: gw3.ano.com
```

```
P-Certificate-Subject-Alternative-Name: gw2.some.com
Max-Forwards: 69
Subject: TBD
Content-Type: application/sdp
Content-Length: 138
Route: <sip:222222@172.16.27.188:5060;lr>
v=0
o=user1 53655765 2353687637 IN IP4 172.16.27.113
s=-
c=IN IP4 172.16.27.113
t=0 0
m=audio 20000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Validating the Request-URI Based on Certificate Information

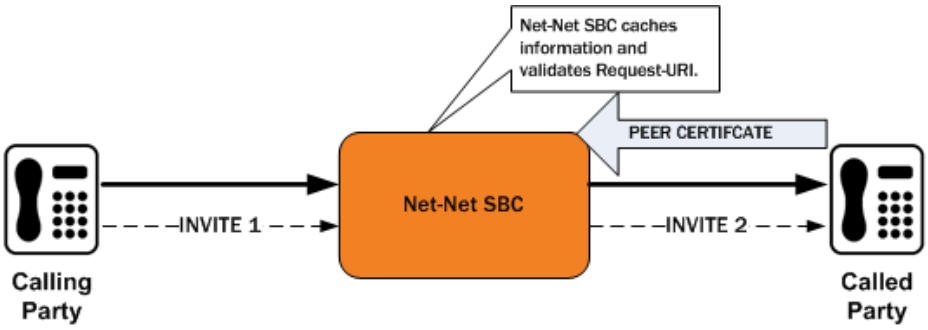
When you configure the Oracle Enterprise Session Border Controller to cache TLS certificate information to validate Request-URIs, it stores the Certificate Subject Name and Certificate Subject Alternative Name (only DNS) it learns from peer certificate attributes. It then takes these actions:

- Extracts the host from the Request-URI of the outgoing INVITE
- Compares this host (exact or wildcard match) with the Certificate Common Name or Certificate Subject Alternative name of the certificate it has received
- Sends out an INVITE if the Certificate Common Name or Certificate Subject Alternative name match; Sends a 403 Forbidden rejection to the endpoint from it received the INVITE if there is no match

Wildcard matching applies only to the prefix of the Request-URI:

```
*.acme.com
    *.*.acmepacket.com
```

This diagram shows a peering scenario where the Oracle Enterprise Session Border Controller receives an INVITE from the calling party, which it processes and prepares to send out INVITE 2. After establishing a TLS connection with the called party and caching the required information, the Oracle Enterprise Session Border Controller validates the Request-URI. Once validation occurs, the Oracle Enterprise Session Border Controller sends INVITE 2.



The peer certificate from the called party during the TLS handshake with the Oracle Enterprise Session Border Controller would look like this. Relevant information in the sample appears in bold font.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9 (0x9)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=MA, L=Woburn, O=Smith Securities, OU=Certificate
Authority Dept, CN=Smith Certificate Authority/emailAddress=amith@CA.com
        Validity
            Not Before: Dec 10 21:14:56 2009 GMT
            Not After : Jul 11 21:14:56 2019 GMT
        Subject: C=US, ST=MA, L=Woburn, O=Acme Packet, OU=Certificate
Authority Dept, CN=*.acme.com/emailAddress=ph2Server@acme.com
        Subject Public Key Info:
```

```
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            X509v3 Issuer Alternative Name:
            email:Smith@CA.com
            X509v3 Subject Alternative Name:
            DNS:gw1.acme.com, DNS:*.ano.com, DNS:*.some.com
            X509v3 Key Usage: critical
            Digital Signature, Key Encipherment
    Signature Algorithm: sha1WithRSAEncryption
```

The outgoing SIP INVITE (INVITE 2 in the diagram) would then look like the sample below. The INVITE is sent because smith.acme.com matches the common name *.acme.com. Other valid SIP Request-URIs would be:

```
222222@gw1.acme.com
222222@smith.ano.com
222222@amith.some.com
```

You can see where the system uses information from the certificate; the text is bold.

```
INVITE sip:222222@smith.acme.com:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.27.113:5060;branch=z9hG4bK4jmg29cmm8l0cg7smmrn85o4q7
From: 111111 <sip:111111@acme.com>;tag=_ph1_tag
To: 222222 <sip:222222@acme.com>
Call-ID: _1-2_call_id-10147@acme.com-1-
CSeq: 1 INVITE
Contact: <sip:111111@172.16.27.113:5060;transport=udp>
Max-Forwards: 69
Subject: TBD
Content-Type: application/sdp
Content-Length: 138
Route: <sip:222222@172.16.27.188:5060;lr>
v=0
o=user1 53655765 2353687637 IN IP4 172.16.27.113
s=-
c=IN IP4 172.16.27.113
t=0 0
m=audio 20000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

TLS Endpoint Certificate Data Caching Configuration

To configure SIP endpoint certificate data caching for an enforcement profile:

1. Access the **enforcement-profile** configuration element.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)# enforcement-profile
   ACMEPACKET(enforcement-profile)#
   ```

2. Select the **enforcement-profile** object to edit.

   ```
   ACMEPACKET(enforcement-profile)# select
   <name>:

   ACMEPACKET(enforcement-profile)#
   ```

3. add-certificate-info—Enter a list of one or more certificate attribute names to enable TLS certificate information caching and insertion of cached certificate information into a customized SIP INVITEs. This parameter is empty by default.

   If you want to list more than one value, enclose the value in quotation marks (“ “) and separate the values with Spaces.

```
ACMEPACKET(enforcement-profile)# add-certificate-info "sub-common-name sub-
alt-name-DNS"
```

4. certificate-ruri-check—Change this parameter from disabled, its default, to enabled if you want your Oracle Enterprise Session Border Controller to cache TLS certificate information and use it to validate Request-URIs. Enabling this parameter also means the Oracle Enterprise Session Border Controller will use the cached TLS certificate information in a customized SIP INVITE.
5. Type **done** to save your configuration.

Untrusted Connection Timeout for TCP and TLS

You can configure the Oracle Enterprise Session Border Controller for protection against starvation attacks for socket-based transport (TCP or TLS) for SIP access applications. During such an occurrence, the attacker would open a large number of TCP/TLS connections on the Oracle Enterprise Session Border Controller and then keep those connections open using SIP messages sent periodically. These SIP messages act as keepalives, and they keep sockets open and consume valuable resources.

Using its ability to promote endpoints to a trusted status, the Oracle Enterprise Session Border Controller now closes TCP/TLS connections for endpoints that do not enter the trusted state within the period of time set for the untrusted connection timeout. The attacking client is thus no longer able to keep connections alive by sending invalid messages.

This feature works by setting a value for the connection timeout, which the Oracle Enterprise Session Border Controller checks whenever a new SIP service socket for TCP or TLS is requested. If the timer's value is greater than zero, then the Oracle Enterprise Session Border Controller starts it. If the timer expires, then the Oracle Enterprise Session Border Controller closes the connection. However, if the endpoint is promoted to the trusted state, then the Oracle Enterprise Session Border Controller will cancel the timer.

Caveats

This connection timeout is intended for access applications only, where one socket is opened per-endpoint. This means that the timeout is not intended for using in peering applications; if this feature were enabled for peering, a single malicious SIP endpoint might cause the connection to be torn down unpredictably for all calls.

Untrusted Connection Timeout Configuration for TCP and TLS

The untrusted connection timer for TCP and TLS is set per SIP interface.

To set the untrusted connection timer for TCP and TLS:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

   If you are adding support for this feature to a pre-existing SIP configuration, then you must select (using the ACLI select command) the configuration that you want to edit.
4. untrusted-conn-timeout—Enter the time in seconds that you want the Oracle Enterprise Session Border Controller to keep TCP and TLS connections open for untrusted endpoints. The default value is 0, which will not start the timer. The valid range is:
   - Minimum—0
   - Maximum—999999999
5. Save and activate your configuration.

Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) is defined in RFC 2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. The protocol enables users to determine the revocation state of a specific certificate, and may provide a more efficient source of revocation information than is possible with Certificate Revocation Lists (CRL).

The protocol specifies the data exchanged between an OCSP client (for example, the Oracle Enterprise Session Border Controller) and an OCSP responder, the Certification Authority (CA), or its delegate, that issued the target certificate. An OCSP client issues a request to an OCSP responder and suspends acceptance of the certificate in question until the responder replies with a certificate status.

Certificate status is reported as

- good
- revoked
- unknown

good indicates a positive response to the status inquiry. At a minimum, this positive response indicates that the certificate is not revoked, but does not necessarily mean that the certificate was ever issued or that the time at which the response was produced is within the certificate's validity interval.

revoked indicates that the certificate has been revoked, either permanently or temporarily.

unknown indicates that the responder cannot identify the certificate.

Caveats

OCSP is currently supported only on TLS interfaces; it is not currently supported for use with IKEv1 and IKEv2.

Online Certificate Status Protocol Configuration

OCSP configuration consists of

1. Configuring one or more certificate status profiles; each profile contains information needed to contact a specific OCSP responder.
2. Enabling certificate revocation checking by assigning a certificate status profile to a previously configured TLS profile.

   To create a certificate status profile:
3. From superuser mode, use the following command sequence to access cert-status-profile configuration mode. While in this mode, you provide the information required to access one or more OCSP responders.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# cert-status-profile
   ACMEPACKET(cert-status-profile)#
   ```
4. Use the required name parameter to identify this cert-status-profile instance — each profile instance provides configuration data for a specific OCSP responder. name is used to distinguish between multiple profile instances.
5. Use the required ip-address parameter to specify the IPv4 address of the OCSP responder.
6. Use the optional port parameter to specify the destination port.

   In the absence of an explicitly configured value, the default port number of 80 is used.
7. Use the optional realm-id parameter to specify the realm used to transmit OCSP requests.

   In the absence of an explicitly configured value, the default specifies service across the wancom0 interface.
8. Use the optional requester-cert parameter only if OCSP requests are signed; ignore this parameter if requests are not signed.

   RFC 2560 does not require signed requests; however, local or CA policies can mandate digital signature..

9. Use the required responder-cert parameter to identify the certificate used to validate OCSP responses — a public key of the OCSP responder.

   RFC 2560 requires that all OCSP responders digitally sign OCSP responses, and that OCSP clients validate incoming signatures.

   Provide the name of the certificate configuration element that contains the certificate used to validate the signed OCSP response.

10. Use the optional retry-count parameter to specify the maximum number of times to retry an OCSP responder in the event of connection failure.

    If the retry counter specified by this parameter is exceeded, the OCSP requester either contacts another responder (if multiple responders have been configured within this cert-status-profile) and quarantine the unavailable responder for a period defined the dead-time parameter.

    In the absence of an explicitly configured value (an integer within the range 0 through 10), the default of 1 is used.

```
ACMEPACKET(cert-status-profile)# retry-count 2
ACMEPACKET(cert-status-profile)#
```

11. Use the optional dead-time parameter to specify the quarantine period imposed on an unavailable OCSP responder.

    In the absence of an explicitly configured value (an integer within the range 0 through 3600 seconds), the default value (0) is used.

    Customer implementations utilizing a single OCSP responder are encouraged to retain the default value, or to specify a brief quarantine period to prevent lengthy service outages.

12. Retain default values for the type and trans-protocol parameter to specify OCSP over an HTTP transport protocol.
13. Use done, exit, and verify-config to complete configuration of this cert-status-profile instance.
14. Repeat Steps 1 through 11 to configure additional certificate status profiles.

    To enable certificate status checking:

15. Move to tls-profile configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)#
```

16. Use the required cert-status-check parameter to enable OCSP in conjunction with an existing TLS profile.
17. Use the required cert-status-profile-list parameter to assign one or more cert-status-profiles to the current TLS profile.

    Each assigned cert-status-profile provides the information needed to access a single OCSP responder.

    Use quotation marks to assign multiple OCSP responders. The following sequence assigns three cert-status-profiles, VerisignClass3Designate, Verisign-1, and Thawte-1 to the TLS-1 profile.

18. Use done, exit, and verify-config to complete configuration.

    Sample Configuration:

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# cert-status-profile
ACMEPACKET(cert-status-profile)# name VerisignClass3Designate
ACMEPACKET(cert-status-profile)# ip-address 192.168.7.100
ACMEPACKET(cert-status-profile)# responder-cert VerisignClass3ValidateOCSP
ACMEPACKET(cert-status-profile)# done
ACMEPACKET(cert-status-profile)# exit
...
...
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tls-profile
```

```
ACMEPACKET(tls-profile)# select
<name>:
1. TLS-1
2. TLS-2
3. TLS-3
selection: 1
ACMEPACKET(tls-profile)# cert-status-check enabled
ACMEPACKET(cert-status-profile)# cert-status-profile-list
VerisignClass3Designate Verisign-1 Thawte-1
ACMEPACKET(cert-status-profile)# done
ACMEPACKET(cert-status-profile)# exit
```

Unreachable OCSR

With OCSP enabled, the client implementation running on the Oracle Enterprise Session Border Controller supports message exchange between the Oracle Enterprise Session Border Controller and an OCSR as specified in RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. The Oracle Enterprise Session Border Controller contacts the OCSR whenever a remote client attempts to establish an SSL/TLS connection with the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller sends a request to the OCSR to check the current status of the certificate presented by the remote client. The Oracle Enterprise Session Border Controller suspends processing of the SSL/TLS connection request pending receipt of the OCSR response. In previous releases (prior to Version S-CX6.3.0), a good OCSR response resulted in the establishment of a secure SSL/TLS connection. A revoked or unknown OCSR response, or the failure to reach an OCSR, resulted in the rejection of the connection attempt.

This behavior, which adheres to the requirements of RFC 2560, conflicts with the requirements of Section 5.4.6.2.1.6.4.a.i of UCR 2008 which requires an OCSP client to attempt authentication of remote clients in the event of an unreachable OCSR.

Release S-CX6.3F1 adds a new attribute (ignore-dead-responder) to the TLS profile configuration element to provide compliance with DISA/DoD requirements specifying OCSP client operations when faced with unreachable OCSRs. By default, the attribute is disabled meaning that all client connections will be disallowed in the event of unreachable OCSRs.

In DISA/DoD environments ignore-dead-responder should be enabled, allowing local certificate-based authentication by the Oracle Enterprise Session Border Controller in the event of unreachable OCSRs. Successful authentication is achieved if the certificate presented by the remote client was signed by a Certificate Authority (CA) referenced by the trusted-ca-certificates attribute. If the local authentication succeeds, the secure TLS/SSL connection is established; otherwise the connection is rejected.

Unreachable OCSR Configuration

The following sample configuration implements DISA/DoD-compliant client behavior in the event of an unreachable OCSR.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security#
ACMEPACKET(security)# tls-profile
ACMEPACKET(security)# show
tls-profile
        name                        DoD
        end-entity-certificate      sylarCert-2048
        trusted-ca-certificates     dod1 dod2 disaA disaB IBM1
        cipher-list                 all
        verify-depth                10
        mutual-authenticate         disabled
        tls-version                 tlsv1
        cert-status-check           enabled
        cert-status-profile-list    DoD
        ignore-dead-responder       enabled
        ...
```

```
        ...
ACMEPACKET(tls-profile)#
```

OCSR Status Monitoring

OCSR monitoring is provided to track the reachability of individual OCSRs, and, in topologies containing multiple OCSRs, the overall availability of OCSR service.

If monitoring is enabled for individual OCSRs, reachability is monitored by observing responder transactions.

Initially, all OCSRs are considered reachable. If a previously reachable OCSR fails to respond to a certificate status request, the Oracle Enterprise Session Border Controller marks the OCSR as unreachable, and generates an SNMP trap and log entry indicating that status. If a previously unreachable OCSR respond to a certificate status request, the Oracle Enterprise Session Border Controller returns the OCSR to the reachable status, and generates an SNMP trap and log entry indicating that status change.

Use the following procedure to enable monitoring of individual OCSRs.

1. Navigate to the new security-config configuration element.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security#
   ACMEPACKET(security)# security-config
   ACMEPACKET(security-config)#
   ```

2. Enable monitoring of individual OCSRs by setting the ocsr-monitoring-traps attribute to enabled; this attribute is disabled by default.

   ```
   ACMEPACKET(security-config)# ocsr-monitoring-traps enabled
   ACMEPACKET(security-config)#
   ```

3. Use done, exit, and verify-config to complete required configuration.

   Reachability status of individual OCSRs is aggregated to monitor the overall availability of OCSR service. Using the procedure explained above, the Oracle Enterprise Session Border Controller maintains a count of all OCSRs, and of all reachable OCSRs.

   - If all OCSRs are reachable, the Oracle Enterprise Session Border Controller generates a trap and log entry noting this optimal state.
   - If all OCSRs are unreachable, the Oracle Enterprise Session Border Controller generates a trap and log entry noting this erroneous state.
   - When the Oracle Enterprise Session Border Controller transitions from either of the two states described above (in the optimal state, when an OCSR becomes unreachable; in the erroneous state, when an unreachable OCSR becomes reachable), the Oracle Enterprise Session Border Controller generates a trap and log entry indicating that an unspecified number of OCSRs are reachable.

   Monitoring of OCSR service availability is a by-product of enabling SNMP; no further configuration is required.

OCSR Access via FQDN

Prior software releases supported OCSR access only via IPv4 addresses and port numbers. In response to a DISA/DoD request, Release S-CX6.3F1 adds support for OCSR access via FQDNs. Since multiple public key infrastructure (PKI) elements capable of supporting OCSP requests can exist within a DISA/DoD environment, the Domain Name Service (DNS) lookup that resolves the FQDN can result in more than one OCSR IP address being returned to the Oracle Enterprise Session Border Controller in its role of OCSP client. When processing a lookup that contains more than one IP address, the Oracle Enterprise Session Border Controller uses a round-robin algorithm to select from the list of OCSR addresses.

OCSR Access via FQDN is available on all media interfaces and on the wancom0 administrative interface. Note that support for FQDN-based access is requires the configuration of DNS support.

If the realm attribute is configured in the certificate-status-profile configuration element, the required DNS query is issued on the corresponding network interface. This model requires configuration of the dns-ip-primary attribute, and optionally the dns-ip-backup1 and dns-ip-backup2 attributes for the realm's network interface.

If the realm attribute is not configured in the certificate-status-profile, the required DNS query is issued on the wancom0 interface. This model requires configuration of the dns-ip-primary attribute, and optionally the dns-ip-backup1 and
dns-ip-backup2 attributes for the wancom0 interface.

Access via an FQDN is supported by a new attribute (hostname) in the
cert-status-profile configuration element.

The Oracle Enterprise Session Border Controller allows configuration of both an OCSR IP address and port number (using the ip-address and port attributes) and an OCSR domain (using the hostname attribute).

In such cases the verify-config command issues a warning and notes that
IP address-based access will be used.

OCSR Access Configuration via IP Address

The following sample configuration accesses an OCSR at 192.168.7.100:8080.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security#
ACMEPACKET(secirity)# cert-status-profile#
ACMEPACKET(cert-status-profile)# show
cert-status-profile
       name                      defaultOCSP
       ip-address                192.168.7.100
       hostname
       port                      8080
       type                      OCSP
       trans-proto               HTTP
       requestor-cert            ocspVerisignClient
       responder-cert            VerisignCA2
       trusted-cas
       realm-id                  admin
       retry-count               1
       dead-time                 0
       last-modified-by
       last-modified-date
ACMEPACKET(cert-status-profile)#
```

OCSR Access Configuration via FQDN

The following sample configuration accesses one or more OCSRs at example.disa.mil.

Note that in the absence of a specified domain, the wancom0 interface must be DNS-enabled.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security#
ACMEPACKET(cert-status-profile)# show
cert-status-profile
        name                        DISAdomain2
        ip-address
        hostname                    example.disa.mil
        port
        type                        OCSP
        trans-proto                 HTTP
        requestor-cert
        responder-cert
        trusted-cas                 dod1 dod2 disaA disaB IBM1
        realm-id
        retry-count                 1
        dead-time                   0
        last-modified-by
        last-modified-date
ACMEPACKET(cert-status-profile)#
```

Direct and Delegated Trust Models

RFC 2560 specifies that an OCSR must digitally sign OCSP responses, and that an OCSP client must validate the received signature. In prior releases, successful validation of the signed response served to authenticate the responder. Such an authentication method is referred to as a direct trust model in that it does not require confirmation from a trusted Certificate Authority (CA). Rather it requires that the OCSP client be in possession of the public counterpart of the private key used by the OCSR to sign the response. This certificate is identified by the responder-cert attribute in the cert-status-profile configuration element. Prior to Release S-CX6.3F1, authentication via signature validation was the only authentication method provided by the OCSP client implementation.

Release S-CX6.3F1 continues support for the direct trust model, while also supporting an alternative delegated trust model as described in Section 5.4.6.2.1.6.1.e.3.c of UCR 2010. The delegated trust model requires that OCSR be authenticated by a trusted CA. Within the DISA/DoD delegated trust model, an OCSR certificate is appended to every response, thus eliminating the need for a pre-provisioned responder certificate. The appended certificate is a signing certificate issued and signed by a DoD-approved CA that issued the certificate that is being validated. These OCSR certificates have a short lifespan and are reissued regularly.

Direct Trust Model Configuration

The direct trust model is used in virtually all commercial/enterprise environments. Configuration of the direct trust model is unchanged from that contained in the latest version of your hardware or the Oracle Enterprise Session Border Controller*Configuration Guide*.

Delegated Trust Model Configuration

The delegated trust model is used exclusively in some strict DISA/DoD environments; other DISA/DoD environments may support both the direct and delegated trust models.

Use the following procedure to configure OCSP for DISA/DoD environments.

1. From superuser mode, use the following command sequence to access cert-status-profile configuration mode. While in this mode, you configure a cert-status-profile configuration element, a container for the information required to access a single, specific OCSR.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# cert-status-profile
   ACMEPACKET(cert-status-profile)#
   ```

2. The name attribute differentiates cert-status-profile configuration elements one from another. Each cert-status-profile provides configuration information for a single, specific OCSP responder.
3. The type attribute selects the certificate revocation check methodology, the only currently supported methodology is OCSP.
4. Retain the default value (http) for trans-protocol attribute, which identifies the transport method used to access the OCSR.
5. The ip-address attribute works in conjunction with the port attribute to provide the IP address of the OCSR.

   ip-address identifies the OCSR by its IP address. port identifies the port monitored by the HTTP server for incoming OCSP requests.

   The port attribute can be safely ignored if the OCSR is specified as a FQDN by the host-name attribute, but is required if the OCSR is identified by the ip-address attribute.

   Allowable port values are integers within the range 1025 through 65535. In the absence of an explicitly configured value, the system provides a default value of 80, the well-known HTTP port.
6. Alternatively, use the host-name attribute to identify the OCSR.

   host-name identifies the OCSR by a FQDN.

   If you provide both an IPv4 address/port number and a FQDN, the Oracle Enterprise Session Border Controller uses the IP address/port number and ignores the FQDN.

If values are provided for both attributes, the Security Gateway uses the IP address and ignores the host-name value.

7. The realm-id attribute specifies the realm used to access the OCSR.

   In the absence of an explicitly configured value, the Oracle Enterprise Session Border Controller provides a default value of wancom0, specifying OCSP transmissions across the wancom0 management interface.

   If the OCSR identified by a FQDN, the realm identified by realm-id must be DNS-enabled.

8. The requester-cert attribute is meaningful only if OCSP requests are signed; ignore this attribute if requests are not signed.

   RFC 2560 does not require the digital signature of OCSP requests. OCSRs, however, can impose signature requirements.

   If a signed request is required by the OCSR, provide the name of the certificate configuration element that contains the certificate used to sign OCSP requests.

9. The responder-cert attribute identifies the certificate used to validate signed OCSP response — a public key of the OCSR.

   In DISA/DoD environments that support the direct trust model, optionally provide the name of the certificate configuration element that contains the certificate used to validate the signed OCSP response.

   If a responder-cert is provided, it is only used if the OCSP response has no appended certificates, in which case the OCSP client attempts to validate the response signature. Depending on the validation failure or success, the response is rejected or accepted.

   If the OCSP response has an appended certificate or certificate chain, the responder-cert is ignored, and the trusted-cas list is used to validate the appended certificate(s).

10. The trusted-cas attribute (a list of certificate configuration objects) identifies the approved DoD-approved CAs that sign OCSR certificates.

    In DISA/DoD environments that support the delegated trust model, you must provide a list of CAs used to validate the received certificate.

    If a certificate or a certificate chain is appended to the OCSP response, the OCSP client verifies that the first certificate signed the response, and that the CA is trusted by the Oracle Enterprise Session Border Controller (that is, the CA certificate is contained in the trusted-cas list. The client then walks through each additional certificate (if any exist) ensuring that each certificate is also trusted. If all certificates are trusted, the OCSP response is accepted; otherwise, it is rejected.

11. The retry-count attribute specifies the maximum number of times to retry an OCSP responder in the event of connection failure.

    If the retry counter specified by this attribute is exceeded, the OCSP requester contacts another responder (if multiple responders have been configured) and quarantines the unavailable responder for a period defined the dead-time attribute.

    In the absence of an explicitly configured value (an integer within the range 0 through 10), the Oracle Enterprise Session Border Controller provides a default value of 1 (connection retries).

12. The dead-time attribute specifies the quarantine period imposed on an unavailable OCSR.

    In the absence of an explicitly configured value (an integer within the range 0 through 3600 seconds), the Oracle Enterprise Session Border Controller provides a default value of 0 (no quarantine period).

    Customer implementations utilizing a single OCSP responder are encouraged to retain the default value, or to specify a brief quarantine period to prevent lengthy service outages.

13. Use done, exit, and verify-config to complete configuration of this cert-status-profile instance.
14. Repeat Steps 1 through 13 to configure additional cert-status-profile configuration elements.

Secure Real-Time Protocol (SRTP) for Software

The Secure Real-Time Transport Protocol, as described in RFC 3711, *The Secure Real-time Transport Protocol (SRTP)*, provides a framework for the encryption and authentication of Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) streams. Both RTP and RTCP are defined by RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*.

Encryption ensures that the call content and associated signalling remains private during transmission.

Authentication ensures the following.

- Received packets are from the purported source
- Packets have not been tampered with during transmission
- A packet has not been replayed by a malicious server

Support for software-based SRTP requires a license key from Oracle.

For instructions to enable a software component with a license key, see "Obtain a License. "

Protocol Overview

While the RFC 3711 framework provides encryption and authentication procedures and defines a set of default cryptographic transforms required for RFC compliance, it does not specify a key management protocol to securely derive and exchange cryptographic keys. RFC 4568, *Session Description Protocol (SDP) Security Description for Media Streams*, defines such a protocol specifically designed to exchange cryptographic material using a newly defined SDP crypto attribute. Cryptographic parameters are established with only a single message or in single round-trip exchange using the offer/answer model defined in RFC 3264, *An Offer/Answer Model with the Session Description Protocol*.

The current release provides support for an initial SDP Security Descriptions (SDES) implementation that generates keys used to encrypt SRTP/SRTCP packets.

Authentication of packets will be added to a subsequent release.

A sample SDP exchange is shown below:

The SDP offerer sends:

```
v=0
o=sam 2890844526 2890842807 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=marge@example.com (Marge Simpson)
c=IN IP4 168.2.17.12
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz|2^20|1:4
```

The SDP answerer replies:

```
v=0
o=jill 25690844 8070842634 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=homer@example.com (Homer Simpson)
c=IN IP4 168.2.17.11
t=2873397526 2873405696
m=audio 32640 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
```

The media-level SDP attribute, crypto, describes the cryptographic suite, key parameters, and session parameters for the preceding unicast media line. The crypto attribute takes the form:

a=crypto: tag crypto-suite key-parameter [session-parameters]

tag

The tag field contains a decimal number that identifies a specific attribute instance. When an offer contains multiple crypto attributes, the answer uses the tag value to identify the accepted offer.

In the sample offer the tag value is 1.

crypto-suite

The crypto-suite field contains the encryption and authentication algorithms, either AES_CM_128_HMAC_SHA1_80 or AES_CM_128_HMAC_SHA1_32.

key-parameter

The key-parameter field contains one or more sets of keying material for the selected crypto-suite and it has following format.

```
"inline:" <key||salt> ["|" lifetime] ["|" MKI ":" length]
```

inline is a method and specifies that the crypto material to be used by the offerer is transmitted via the SDP.

The key||salt field contains a base64-encoded concatenated master key and salt.

Assuming the offer is accepted, the key || salt provides the crypto material used by the offerer to encrypt SRTP/SRTCP packets, and used by the answerer to decrypt SRTP/SRTCP packets.

Conversely the key || salt contained in the answer to the offer provides the crypto material used by the answerer to encrypt SRTP/SRTCP packets, and used by the offerer to decrypt SRTP/SRTCP packets.

The lifetime field optionally contains the master key lifetime (maximum number of SRTP or SRTCP packets encoded using this master key).

In the sample offer the lifetime value is 1,048, 576 (220) packets.

The MKI:length field optionally contains the Master Key Index (MKI) value and the MKI length.

The MKI is used only when the offer contains multiple keys; it provides a means to differentiate one key from another. The MKI takes the form of an integer, followed by its byte length when included in SRTP/SRTCP packets.

In the sample offer the MKI value is 1 with a length of 4 bytes.

The session-parameters field contains a set of optional parameters that may override SRTP session defaults for the SRTP and SRTCP streams.

UNENCRYPTED_SRTP — SRTP messages are not encrypted

UNENCRYPTED_SRTCP — SRTCP messages are not encrypted

UNAUTHENTICATED_SRTP — SRTP messages are not authenticated

When generating an initial offer, the offerer ensures that there is at least one crypto attribute for each media stream for which security is desired. Each crypto attribute for a given media stream must contain a unique tag. The ordering of multiple crypto attributes is significant — the most preferred crypto suite is listed first.

Upon receiving the initial offer, the answerer must either accept one of the offered crypto attributes, or reject the offer in its entirety.

When an offered crypto attribute is accepted, the crypto attribute contained in the answer MUST contain the tag and crypto-suite from the accepted crypto attribute in the offer, and the key(s) the answerer will use to encrypt media sent to the offerer.

The crypto-suite is bidirectional and specifies encryption and authentication algorithms for both ends of the connection. The keys are unidirectional in that one key or key set encrypts and decrypts traffic originated by the offerer, while the other key or key set encrypts and decrypts traffic originated by the answerer.

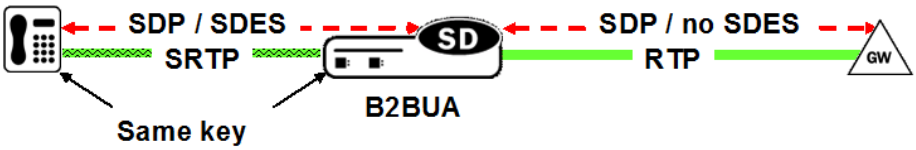
Key exchange via text-based SDP is unacceptable in that malicious network elements could easily eavesdrop and obtain the plaintext keys, thus compromising the privacy and integrity of the encrypted media stream. Consequently, the SDP exchange must be protected by a security protocol such as TLS.

Operational Modes

SRTP topologies can be reduced to three basic topologies which are described in the following sections.

Single-Ended SRTP Termination

Single-ended SRTP termination is illustrated in the following figure.



Single-Ended SRTP Termination

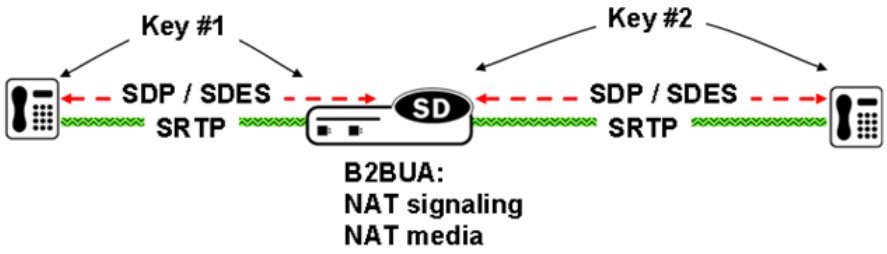
If SRTP is enabled for the inbound realm/interface, the Oracle Enterprise Session Border Controller handles the incoming call as specified by the Media Security Policy assigned to the inbound realm. If there is crypto attribute contained in the offer, the Oracle Enterprise Session Border Controller parses the crypto attributes and optional parameters, if any. If the offer contains a crypto attribute or attributes compatible with the requirements specified by the SDES profile assigned to the Media Security policy, it selects the most preferred compatible attribute. Otherwise, the Oracle Enterprise Session Border Controller rejects the offer. Before the SDP is forwarded to the called party, the Oracle Enterprise Session Border Controller allocates resources, established SRTP and SRTCP Security Associations and updates the SDP by removing the crypto attribute and inserting possibly NAT'ed media addresses and ports. At the same time, the original crypto attribute is also removed from the SDP.

Once the reply from the called party is received, the Oracle Enterprise Session Border Controller inserts appropriate crypto attribute(s) to form a new SDP, and forward the response back to the calling party.

Refer to *ACLI Example Configurations* for a sample ACLI configuration.

Back-to-Back SRTP Termination

Back-to-back SRTP termination is illustrated in the following figure.



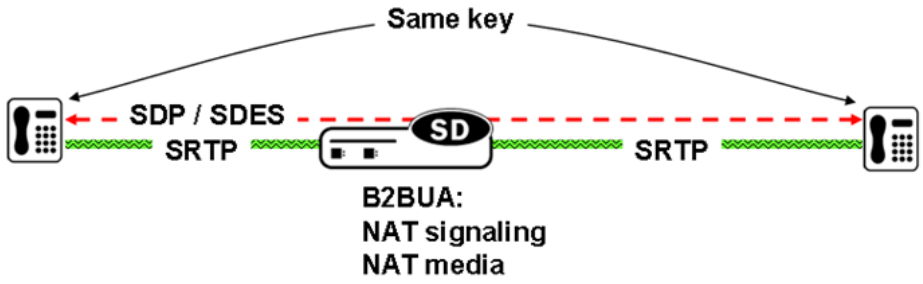
Back-to-Back SRTP Termination

Initial processing is similar to the single-ended termination described above. Before forwarding the request to the called party, the Net-Net ESD replaces the original crypto attribute with a new one whose crypto attribute conforms to the media security policy for the outbound realm/interface. Upon receiving the answer from the called party, the Net-Net ESD accepts or rejects it, again based upon conformity to the media security policy. If accepted, the Oracle Enterprise Session Border Controller replaces the original crypto attribute from the called party with its own to form a new SDP, which it forwards back to the calling party. At this point, SRTP media sessions are established on both sides for both calling and called parties.

Refer to *ACLI Example Configurations* for a sample ACLI configuration.

SRTP Pass-Thru

SRTP pass-thru is illustrated in the following figure.



SRTP Pass-Thru

If the media security policy specifies pass-through mode, the Net-Net ESD does not alter the crypto attribute exchange between the calling and the called party; the attribute is transparently passed.

Refer to *ACLI Example Configurations* for a sample ACLI configuration.

ACLI Instructions

SDES configuration consists of the following steps.

1. Create one or more SDES profiles which specify parameter values negotiated during the offer/answer exchange.
2. Create one or more Media Security Policies that specify key exchange protocols and protocol-specific profiles.
3. Assign a Media Security Policy to a realm.
4. Create an interface-specific Security Policy (refer to *Security Policy* for a sample ACLI configuration)

SDES Profile Configuration

An SDES profile specifies the parameter values offered or accepted during SDES negotiation.

To configure SDES profile parameters:

1. From superuser mode, use the following command sequence to access sdes-profile configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# sdes-profile
ACMEPACKET(sdes-profile)#
```

2. Use the required name parameter to provide a unique identifier for this sdes-profile instance.

   name enables the creation of multiple sdes-profile instances.
3. Use the crypto-suite parameter to select the algorithms accepted or offered by this sdes-profile.

   Note: SRTP authentication is not currently supported.

   Allowable values are:

   AES_CM_128_HMAC_SHA1_80 (the default value)

   supports AES/128 bit key for encryption and HMAC/SHA-1 80-bit digest for authentication

   AES_CM_128_HMAC_SHA1_32

   supports AES/128 bit key for encryption and HMAC/SHA-1 32-bit digest for authentication
4. Because SRTP authentication is not currently supported, ignore the srtp-auth parameter.
5. Use the srtp-encrypt parameter to enable or disable the encryption of RTP packets.

With encryption enabled, the default condition, the Oracle Enterprise Session Border Controller offers RTP encryption, and rejects an answer that contains an UNENCRYPTED_SRTP session parameter in the crypto attribute.

With encryption disabled, the Oracle Enterprise Session Border Controller does not offer RTP encryption and includes an UNENCRYPTED_SRTP session parameter in the SDP crypto attribute; it accepts an answer that contains an UNENCRYPTED_SRTP session parameter.

6. Use the srtcp-encrypt parameter to enable or disable the encryption of RTCP packets.

   With encryption enabled, the default condition, the Oracle Enterprise Session Border Controller offers RTCP encryption, and rejects an answer that contains an UNENCRYPTED_SRTCP session parameter in the crypto attribute.

   With encryption disabled, the Oracle Enterprise Session Border Controller does not offer RTCP encryption and includes an UNENCRYPTED_SRTCP session parameter in the SDP crypto attribute; it accepts an answer that contains an UNENCRYPTED_SRTCP session parameter.

7. Use the key and salt parameters to generate the synchronous key used to encrypt and decrypt SRTP/SRTCP traffic originated by the Net-Net ESD. These concatentated values are passed to the remote SRTP peer as described in *Protocol Overview*. Upon reception, the remote peer imputs the key and salt values to the negotiated encryption algorithm (AES in the current implmentation), thus deriving the key required to decrypt SRTP/SRTCP traffic received from the Oracle Enterprise Session Border Controller.

   The key parameter provides the basic keying material, while the salt (a bit string) provides the randomsess/entropy required by the encryption algorithm.

8. Use the mki parameter to enable or disable the inclusion of the MKI:length field in the SDP crypto attribute.

   The master key identifier (MKI) is an optional field within the SDP crypto attribute that differentiates one key from another. MKI is expressed as a pair of decimal numbers in the form: |mki:mki_length| where mki is the MKI integer value and mki_length is the length of the MKI field in bytes.

   The MKI field is necessary only if the SDES offer contains multiple keys within the crypto attribute.

   Allowable values are enabled and disabled (the default).

   enabled – an MKI field is sent within the crypto attribute (16 bytes maximum)

   disabled – no MKI field is sent

9. Use the egress-offer-format to specify the egress offer format for this profile to use when you set the outbound mode in the associated media security policy to any (refer to *Media Security Policy Configuration*. You can select one of two values:

   If the media security policy requires the use of either RTP or SRTP, this paramater can be safely ignored. If the media security policy is permissive (the mode parameter is set to any), select one of the two supported values.

   - same-as-ingress (default), the Oracle Enterprise Session Border Controller leaves the profile of the media lines unchanged
   - simultaneous-best-effort, the Oracle Enterprise Session Border Controller inspects the incoming offer SDP and:
     - Adds an RTP/SAVP media line for any media profile that has only the RTP/AVP media profile
     - Adds an RTP/AVP media line for any media profile that has only the RTP/SAVP media profile

10. Use done, exit, and verify-config to complete configuration of this SDES profile instance.
11. Repeat Steps 1 through 8 to configure additional SDES profiles.

Media Security Policy Configuration

Use the following procedure to create a Media Security Policy that specifies the role of the Oracle Enterprise Session Border Controller in the security negotiation. If the Oracle Enterprise Session Border Controller takes part in the negotiation, the policy specifies a key exchange protocol and SDES profile for both incoming and outgoing calls.

To configure media-security-policy parameters:

1. From superuser mode, use the following command sequence to access media-sec-policy configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# media-sec-policy
ACMEPACKET(media-sec-policy)#
```

2. Use the required name parameter to provide a unique identifier for this media-sec-policy instance.

   name enables the creation of multiple media-sec-policy instances.

3. Use optional pass-through parameter to enable or disable pass-thru mode.

   With pass-through mode disabled (the default state), the Net-Net ESD disallows end-to-end negotiation — rather the Oracle Enterprise Session Border Controller initiates and terminates SRTP connections with both endpoints.

   With pass-through mode enabled, the SRTP endpoints negotiate security parameters between each other; consequently, the Oracle Enterprise Session Border Controller simply relays SRTP traffic between the two endpoints.

4. Use the outbound navigation command to move to media-sec-outbound configuration mode. While in this configuration mode you specify security parameters applied to the outbound call leg, that is calls sent by the Oracle Enterprise Session Border Controller.
5. Use the profile parameter to specify the name of the SDES profile applied to calls sent by the Oracle Enterprise Session Border Controller.
6. Use the mode parameter to select the real time transport protocol.

   Allowable values are rtp (the default) | srtp | any (either rtp | srtp)

   mode identifies the transport protocol (RTP or SRTP) included in an SDP offer when this media-security-policy is in effect.

7. Use the protocol parameter to select the key exchange protocol.

   Select sdes for SDES key exchange.

8. Use the done and exit parameters to return to media-sec-policy configuration mode.
9. Use the inbound navigation command to move to media-sec-inbound configuration mode. While in this configuration mode you specify security parameters applied to the inbound call leg, that is calls received by the Oracle Enterprise Session Border Controller.
10. Use the profile parameter to specify the name of the SDES profile applied to calls received by the Oracle Enterprise Session Border Controller.
11. Use the mode parameter to select the real time transport protocol.

    Allowable values are rtp (the default) | srtp | any (either rtp | srtp)

    mode identifies the transport protocol (RTP or SRTP) included in an SDP offer when this media-security-policy is in effect.

12. Use the protocol parameter to select the key exchange protocol.

    Select sdes for SDES key exchange.

13. Use done, exit, and verify-config to complete configuration of this media security policy instance.
14. Repeat Steps 1 through 13 to configure additional media-security policies.

Assign the Media Security Policy to a Realm

To assign a media-security-policy to a realm:

1. From superuser mode, use the following command sequence to access realm-config configuration mode. While in this mode, you assign an existing media-security-policy to an existing realm.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)# select
```

```
identifier:
1. access-12
...
...
selection: 1
ACMEPACKET(realm-config)#
```

2. Use the media-sec-policy parameter to assign the policy to the target realm.
3. Use done, exit, and verify-config to complete assignment of the media-security-policy to the realm.

ACLI Example Configurations

The following section contain XML representations of system configurations for basic operational modes.

Single-Ended SRTP Termination Configuration

```
<sdesProfile name='sdes'
    srtpAuth='enabled'
    srtpEncrypt='enabled'
    srtcpEncrypt='enabled'
    mki='disabled'
    egressOfferFormat='same-as-ingress'
    useIngressSessionParams=''
    options=''
    key=''
    salt=''
    lastModifiedBy='admin@172.30.11.55'
    lastModifiedDate='2013-03-04 19:29:40' objectId='21'>
    <cipherSuites name='AES_CM_128_HMAC_SHA1_80'/>
    key>sdes</key>
</sdesProfile>

<mediaSecPolicy name='sdes'
    passThrough='disabled'
    options=''
    lastModifiedBy='admin@172.30.11.55'
    lastModifiedDate='2013-03-04 19:30:23' objectId='22'>

    <inbound profile='sdes'
        mode='srtp'
        protocol='sdes'/>

    <outbound profile='sdes'
        mode='sdes'
        protocol='sdes'/>

    key>sdes</key>
        </mediaSecPolicy>

...
...
...
realm-config
    identifier                peer
    description
    addr-prefix               192.168.0.0/16
    network-interfaces        M00:0
    mm-in-realm               enabled
    mm-in-network             enabled
    mm-same-ip                enabled
    mm-in-system              enabled
    bw-cac-non-mm             disabled
    msm-release               disabled
    qos-enable                disabled
```

```
    generate-UDP-checksum     disabled
    max-bandwidth             0
    fallback-bandwidth        0
    max-priority-bandwidth    0
    max-latency               0
    max-jitter                0
    max-packet-loss           0
    observ-window-size        0
    parent-realm
    dns-realm
    media-policy
    media-sec-policy          msp2
    in-translationid
    ...
    ...
    ...
    last-modified-by         admin@console
    last-modified-date       2009-11-10 15:38:19
```

Back-to-Back SRTP Termination Configuration

```
ACMEPACKET# show running-config
...
...
...
sdes-profile
     name                           sdes1
     crypto-list                    AES_CM_128_HMAC_SHA1_80
     srtp-auth                      enabled
     srtp-encrypt                   enabled
     srtcp-encrypt                  enabled
     mki                            disabled
     key
     salt
     last-modified-by               admin@console
     last-modified-date             2009-11-16 15:37:13
media-sec-policy
     name                           msp2
     pass-through                   disabled
     inbound
          profile                   sdes1
     mode                           srtp
     protocol                       sdes
     outbound
          profile                   sdes1
     mode                           srtp
     protocol                       sdes
     last-modified-by               admin@console
     last-modified-date             2009-11-16 15:37:51
...
...
...
realm-config
     identifier                     peer
     description
     addr-prefix                    192.168.0.0/16
     network-interfaces    M00:0
     mm-in-realm    enabled
     mm-in-network                  enabled
     mm-same-ip                     enabled
     mm-in-system                   enabled
     bw-cac-non-mm                  disabled
     msm-release                    disabled
     qos-enable                     disabled
```

```
     generate-UDP-checksum          disabled
     max-bandwidth                  0
     fallback-bandwidth             0
     max-priority-bandwidth         0
     max-latency                    0
     max-jitter                     0
     max-packet-loss                0
     observ-window-size             0
     parent-realm
     dns-realm
     media-policy
     media-sec-policy               msp2
     in-translationid
     ...
     ...
     ...
realm-config
     identifier                     backOffice
     description
     addr-prefix                       172.16.0.0/16
     network-interfaces    M10:0
     mm-in-realm                    enabled
     mm-in-network                  enabled
     mm-same-ip                     enabled
     mm-in-system                   enabled
     bw-cac-non-mm                  disabled
     msm-release                    disabled
     qos-enable                     disabled
     generate-UDP-checksum          disabled
     max-bandwidth                  0
     fallback-bandwidth             0
     max-priority-bandwidth         0
     max-latency                    0
     max-jitter                     0
     max-packet-loss                0
     observ-window-size             0
     parent-realm
     dns-realm
     media-policy
     media-sec-policy               msp2
     in-translationid
     ...
     ...
     ...
     last-modified-by              admin@console
     last-modified-date            2009-11-10 15:38:19
```

SRTP Pass-Thru Configuration

```
ACMEPACKET# show running-config
...
...
...
sdes-profile
     name                           sdes1
     crypto-list                    AES_CM_128_HMAC_SHA1_80
     srtp-auth                      enabled
     srtp-encrypt                   enabled
     srtcp-encrypt                  enabled
     mki                            disabled
     key
     salt
     last-modified-by               admin@console
     last-modified-date             2009-11-16 15:37:13
```

```
     media-sec-policy
     name                            msp2
     pass-through                    enabled
     inbound
     profile                         sdes1
     mode                            srtp
     protocol                        sdes
     outbound
     profile                         sdes1
     mode                            srtp
     protocol                        sdes
     last-modified-by                admin@console
     last-modified-date              2009-11-16 15:37:51
     ...
     ...
     ...
realm-config
     identifier                      peer
     description
     addr-prefix                     192.168.0.0/16
     network-interfaces              M00:0
     mm-in-realm                     enabled
     mm-in-network                   enabled
     mm-same-ip                      enabled
     mm-in-system                    enabled
     bw-cac-non-mm                   disabled
     msm-release                     disabled
     qos-enable                      disabled
     generate-UDP-checksum           disabled
     max-bandwidth                   0
     fallback-bandwidth              0
     max-priority-bandwidth          0
     max-latency                     0
     max-jitter                      0
     max-packet-loss                 0
     observ-window-size              0
     parent-realm
     dns-realm
     media-policy
     media-sec-policy                msp2
     ...
     ...
     ...
realm-config
     identifier                      core
     description
     addr-prefix                     172.16.0.0/16
     network-interfaces              M10:0
     mm-in-realm                     enabled
     mm-in-network                   enabled
     mm-same-ip                      enabled
     mm-in-system                    enabled
     bw-cac-non-mm                   disabled
     msm-release                     disabled
     qos-enable                      disabled
     generate-UDP-checksum           disabled
     max-bandwidth                   0
     fallback-bandwidth              0
     max-priority-bandwidth          0
     max-latency                     0
     max-jitter                      0
     max-packet-loss                 0
     observ-window-size              0
     parent-realm
```

```
     dns-realm
     media-policy
     media-sec-policy               msp2
     in-translationid
     ...
     ...
     ...
     last-modified-by               admin@console
     last-modified-date             2009-11-10 15:38:19
```

Security Policy

A Security Policy enables the Oracle Enterprise Session Border Controller to identify inbound and outbound media streams that are treated as SRTP/SRTCP. The high-priority Security Policy, p1, (shown below) allows signaling traffic from source 172.16.1.3 to destination 172.16.1.10:5060. The lower-priority Security Policy, p2, (also shown below) matches media traffic with the same source and destination, but without any specific ports. Consequently, SIP signaling traffic (from local port 5060) go through, but the media stream will be handled by appropriate SRTP SA.

```
security-policy
     name                           p1
     network-interface              private:0
     priority                       0
     local-ip-addr-match            172.16.1.3
     remote-ip-addr-match           172.16.1.10
     local-port-match               5060
     remote-port-match              0
     trans-protocol-match           UDP
     direction                      both
     local-ip-mask                  255.255.255.255
     remote-ip-mask                 255.255.255.255
     action                         allow
     ike-sainfo-name
     outbound-sa-fine-grained-mask
          local-ip-mask             255.255.255.255
          remote-ip-mask            255.255.255.255
          local-port-mask           0
          remote-port-mask          0
          trans-protocol-mask       0
          valid                     enabled
          vlan-mask                 0xFFF
     last-modified-by               admin@console
     last-modified-date             2009-11-09 15:01:55

     security-policy
     name                           p2
     network-interface              private:0
     priority                       10
     local-ip-addr-match            172.16.1.3
     remote-ip-addr-match           172.16.1.10
     local-port-match               0
     remote-port-match              0
     trans-protocol-match           UDP
     direction                      both
     local-ip-mask                  255.255.255.255
     remote-ip-mask                 255.255.255.255
     action                         srtp
     ike-sainfo-name
     outbound-sa-fine-grained-mask
     local-ip-mask                  0.0.0.0
     remote-ip-mask                 255.255.255.255
     local-port-mask                0
     remote-port-mask               65535
     trans-protocol-mask            255
     valid                          enabled
```

```
    vlan-mask                     0xFFF
    last-modified-by              admin@console
    last-modified-date            2009-11-09 15:38:19
```

Modified ALCI Configuration Elements

The action parameter in security-policy configuration mode has been modified to accept additional values, srtp and srtcp.

From superuser mode, use the following command sequence to access media-sec-policy configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)# security-policy
ACMEPACKET(security-policy)# action ?
<enumeration> action (default: ipsec)
ipsec, allow, discard, srtp, srtcp
ACMEPACKET(security-policy)#
```

Refer to *Security Policy* for sample Security Policies.

The show security command has been updated with an srtp option.

```
ACMEPACKET# show security srtp
sad
spd
statistics
SRTP Statistics
status
```

The srtp option is similar to the ipsec option save for the sad sub-option that provides data for only SRTP SAs.

The show sa stats command has been updated with an srtp option.

```
ACMEPACKET# show sa stats <ENTER>    Show statistics summary of all
Security Associations
<ike>       Show statistics for IKE Security Associations
<ims-aka>   Show statistics for IMS-AKA Security Associations
<srtp>      Show statistics for SRTP Security Associations
sd# show sa stats srtp
20:06:24-114
SA Statistics                   ---- Lifetime ----
                             Recent     Total      PerMax
SRTP Statistics
ADD-SA Req Rcvd              0          0          0
ADD-SA Success Resp Sent     0          0          0
ADD-SA Fail Resp Sent        0          0          0
DEL-SA Req Rcvd              0          0          0
DEL-SA Success Resp Sent     0          0          0
DEL-SA Fail Resp Sent        0          0          0
```

ARIA Cipher Support

Previous and the current Oracle Enterprise Session Border Controller releases have provided support for the Secure Real-Time Transport Protocol (SRTP), as defined in RFC 3711, to encrypt and authenticate Real-Time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP) media streams. Concurrent support for Session Description Protocol Security Descriptions (SDES) and Multimedia Internet Keying (MIKEY) enabled the exchange of SRTP keying material. These releases have supported a single encryption algorithm, Advanced Encryption System (AES) counter mod with 128-bit keys

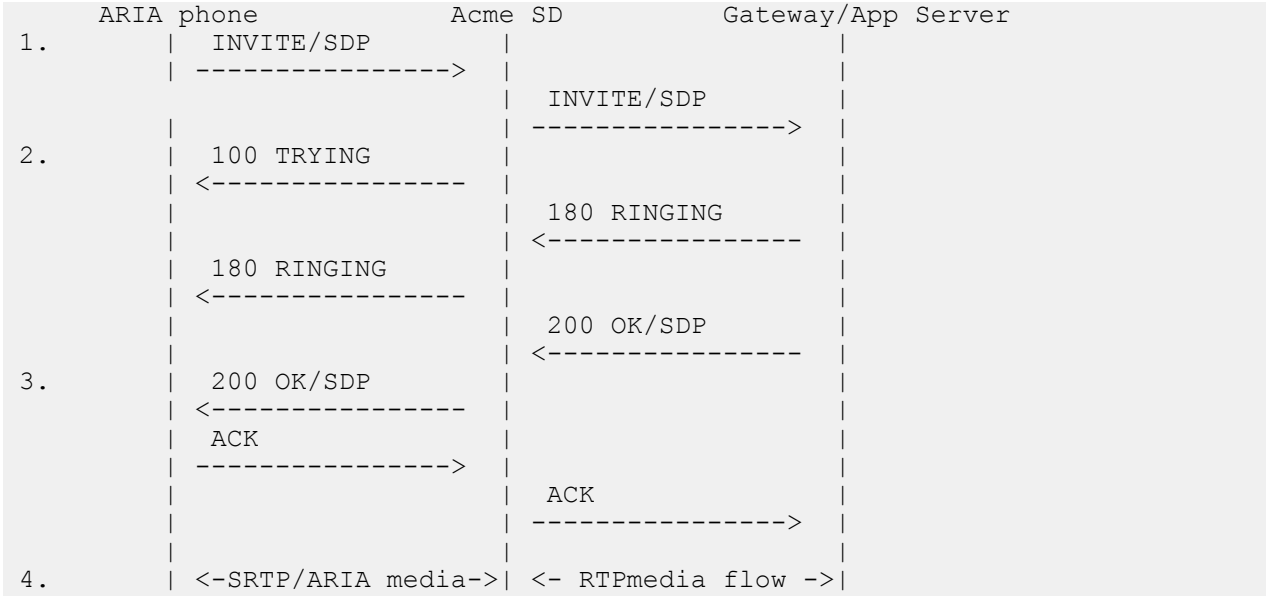
This release supports ARIA, a block cipher selected by the Korean Agency for Technology and Standards as a standard cryptographic Technique. The Oracle Oracle Enterprise Session Border Controller now supports the ARIA cipher with a 192-bit key in counter mode for RTP and RTCP encryption; authentication is supported by HMAC_SHA1 with either 32-bit or 80-bit keys.

Hardware Requirements

ARIA support requires the ETC (Enhanced Traffic Controller) Network Interface Unit (NIU).

Call Flow

An example call flow between a ARIA endpoint, the OracleSD, and a Gateway/Application Server illustrates a successful call establishment where the call is originated from an ARIA enabled phone and destined to a core network server.



1. The ARIA-enabled phone sends an INVITE request to the SD with the crypto attribute in the SDP specifying the ARIA 192 CM cipher for encryption and HMAC_SHA1_80 for authentication. The crypto attribute also has the master key encoded in base-64 format, as well the mki parameters (optionally). The SD forwards the INVITE to the called party via the gateway according to the media-security-policy on the outbound realm.
2. The SD sends provisional response to INVITE request
3. Assuming that the SD gets successful answer from called party, the SD sends a 200 OK response to the caller, with the crypto attribute in the accompanying SDP specifying the ARIA 192 CM cipher for encryption and HMAC_SHA1_80 for authentication. The crypto attribute also has the master key, as well the mki parameters (optionally).
4. The ARIA-enabled phone acknowledges the reception of 200 OK final response. At this point, encrypted SRTP traffic using the ARIA 192 counter mode cipher flows between the phone and the SD, and unencrypted traffic flows between the SD and the core network.

ARIA Support Configuration

ARIA support is enabled at the sdes-profile level.

1. Use the following command sequence to move to sdes-profile Configuration Mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# media-security
ACMEPACKET(sdes-profile)#
```

2. Use the crypto-list parameter to specify the crypto suite used for SDES-based encryption.

   Use either aria_cm_192_hmac_32, or aria_cm_192_hmac_32 to specify ARIA encryption.

```
ACMEPACKET(sdes-profile)# crypto-list aria_cm_192_hmac_80
ACMEPACKET(sdes-profile)#
```

3. Use done, exit, and verify-config to complete cipher suite selection.

Secure and Non-Secure Flows in the Same Realm

To simplify deployments, the E-SBC allows secure and non-secure flows in the same realm.This broadened set of capabilities means the E-SBC can support RTP and SRTP flows, and it can support a larger group of UAs that might have varying SRTP abilities. Prior to this release, when a cryptographic session arrived at the E-SBC and failed to match an applicable media security profile, it was rejected.

This broadened support for secure and non-secure flows and for UAs with various SRTP abilities is established throughout the OS, residing in these configurations:

- media-sec-policy
- sdes-profile

While configurations reside there, you should also note special considerations for the security-policy configuration and implications for security associations.

Mode Settings in the Media Security Policy

The media security policy configuration's mode parameter offers three settings. It is the any mode that allows you to support secure and non-secure flows in the same realm.

For Incoming Flows

This section describes the way all three settings behavior for incoming flows.

- rtp—If the incoming media security policy associated with a realm has rtp set as its mode, then the E-SBC only accepts offer SDP containing RTP/AVP media lines. Otherwise, the E-SBC rejects the session with a 488 Not Acceptable Here.
- srtp—If the incoming media security policy associated with a realm has srtp set as its mode, the E-SBC only accepts offer SDP containing RTP/SAVP media lines. Otherwise, the E-SBC rejects the session with a 488 Not Acceptable Here.
- any—If the incoming media security policy associated with a realm has any set as its mode, the E-SBC accepts offer SDP that has RTP/AVP media lines, RTP/SAVP media lines, or both.

For Outgoing Flows

This section describes the way all three settings behavior for outgoing flows.

- rtp—If the outgoing media security policy associated with a realm has rtp set as its mode, then the E-SBC converts any RTP/SAVP media lines from incoming offer SDP to RTP/AVP for the offer SDP it sends out.

  Incoming offer SDP might look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 192.168.22.180
s=SIP Call
c=IN IP4 192.168.22.180
t=0 0
m=audio 5010 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
a=crypto:0 AES_CM_128_HMAC_SHA1_80 inline:f0oLKTuMYwXqrKa7Ch
+MOBvLe8YnXnD6Kmnj4LQ2
```

  The E-SBC will take that and convert it to the following for outgoing traffic.

```
v=0
o=MxSIP 0 1480968866 IN IP4 172.16.22.180
s=SIP Call
c=IN IP4 172.16.22.180
t=0 0
m=audio 6000 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-1
```

This conversion can result in multiple media lines with RTP/AVP for the same media profile and an RTP/SAVP media line for the same media profile. To prevent duplicate lines in the SDP the E-SBC sends, the E-SBC inspects incoming SDP to determine is RTP/AVP and RTP/SAVP media lines exist for the same media profile. If it finds such a media profile, the E-SBC disables the RTP/AVP (by setting the port to 0 in the outgoing offer SDP) corresponding to the RTP/AVP media line for that media profile. Doing so forces the UA answering the SDP offer to choose the media lines corresponding to the RTP/SAVP media lines in the incoming offer SDP. An SRTP-RTP session results.

The incoming offer SDP might look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 192.168.22.180
s=SIP Call
c=IN IP4 192.168.22.180
t=0 0
m=audio 5012 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
m=audio 5010 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
a=crypto:0 AES_CM_128_HMAC_SHA1_80 inline:f0oLKTuMYwXqrKa7Ch
+MOBvLe8YnXnD6Kmnj4LQ2
```

And the outgoing offer SDP will look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 172.16.22.180
s=SIP Call
c=IN IP4 172.16.22.180
t=0 0
m=audio 0 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
```

```
a=fmtp:18 annexb=no
a=fmtp:101 0-15
m=audio 6002 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
```

- srtp—If the outgoing media security policy associated with a realm has srtp set as its mode, the E-SBC converts any RTP/AVP media lines from an incoming offer SDP to RTP/SAVP for the offer SDP the E-SBC sends.

  The incoming offer SDP might look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 192.168.22.180
s=SIP Call
c=IN IP4 192.168.22.180
t=0 0
m=audio 5012 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
```

  And the outgoing offer SDP will look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 172.16.22.180
s=SIP Call
c=IN IP4 172.16.22.180
t=0 0
m=audio 6000 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-1
a=crypto:0 AES_CM_128_HMAC_SHA1_80 inline:f0oLKTuMYwXqrKa7Ch
+MOBvLe8YnXnD6Kmnj4LQ2
```

  This conversion might result in multiple media lines with RTP/SAVP for the same media profile if the incoming offer SDP has an RTP/AVP media line and an RTP/SAVP media for the same media profile. To prevent multiple identical media lines in the SDP it sends, the E-SBC inspects the incoming SDP to determine whether both RTP/AVP and RTP/SAVP media lines exist for the same media profile. If it finds such a media profile, the E-SBC disables the RTP/SAVP (by setting the port to 0 in the outgoing offer SDP) corresponding to the RTP/AVP media line for that media profile. Doing so forces the UA answering the SDP offer to choose the media lines corresponding to the RTP/SAVP media lines in the incoming offer SDP. An SRTP-SRTP session results.

  The incoming offer SDP might look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 192.168.22.180
s=SIP Call
c=IN IP4 192.168.22.180
t=0 0
```

```
m=audio 5012 RTP/AVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
m=audio 5010 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
a=crypto:0 AES_CM_128_HMAC_SHA1_80 inline:f0oLKTuMYwXqrKa7Ch
+MOBvLe8YnXnD6Kmnj4LQ2
```

And the outgoing offer SDP will look like this:

```
v=0
o=MxSIP 0 1480968866 IN IP4 172.16.22.180
s=SIP Call
c=IN IP4 172.16.22.180
t=0 0
m=audio 0 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-15
m=audio 6002 RTP/SAVP 0 8 18 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=fmtp:18 annexb=no
a=fmtp:101 0-1
a=crypto:0 AES_CM_128_HMAC_SHA1_80 inline:f0oLKTuMYwXqrKa7Ch
+MOBvLe8YnXnD6Kmnj4LQ2
```

- any—If the outgoing media security policy associated with a realm has any set as its mode, the E-SBC creates offer SDP based on the value configured in the egress-offer-format, which is set either in the sdes-profile configuration.
  - If the value is same-as-ingress, the E-SBC leaves the profile of the media lines unchanged.
  - If the value is simultaneous-best-effort, the E-SBC inspects the incoming offer SDP and:
    - Adds an RTP/SAVP media line for any media profile that has only the RTP/AVP media profile
    - Adds an RTP/AVP media line for any media profile that has only the RTP/SAVP media profile

    Should the media profile in the incoming offer SDP already have two media lines (one fore RTP/AVP and on for RTP/SAVP), the E-SBC does not have to make these additions. It will map the media lines in the answer it receives with the media lines from the incoming offer SDP. It will also ensure that media lines in the answer SDP it sends match the media lines from the incoming offer SDP.

Using Security Associations for RTP and RTCP

With RTP and SRTP supported in the same realm, you want to configure your SRTP security policies to preserve system resources and exercise the full capability number of licensed sessions. You need to do to avoid session agent interaction that can have an adverse impact on the number of sessions.

To do so, check the local-ip-match-address for the STRP security policy has an IP address different from the all steering pool IP addresses for realms requiring both RTP and SRTP. The E-SBC recognizes this difference automatically and sets the connection address of media lines in SDP accordingly:

- The connection address for RTP media lines is the IP address of the applicable steering pool. The E-SBC passes through RTP and RTCP packets sent by and received from the steering pool IP address. This operation requires no reference to session agents because the steering pool address does not match the IP address for the SRTP security policy's local-ip-address-match value.
- The connection address of the SRTP media lines continues to be the local-ip-address-match value from the applicable SRTP security policy.

Since RTP and RTCP packets are sent to and from the steering pool's IP address (an IP address for which there is no SRTP security policy configured), there is no reason to reference session agents.

Note: Oracle 's Enhanced Traffic Controller (ETC) networking interface unit handles traffic differently such the issue with session agent reference is elided. That is, if you are using the ETC NIU (available with OS Release S-CX6.3.0 and later), you do not need to be concerned about this issue.

ACLI Instructions and Examples

This section shows you how to configure your E-SBC to support secure and non-secure flows in the same realm.

To configure a security policy to support secure and non-secure flows in the same realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type media-security and press Enter.

```
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)#
```

4. Type media-sec-policy and press Enter. If you are editing a pre-existing configuration, you needs to select it before you can make changes.

```
ACMEPACKET(media-security)# media-sec-policy
ACMEPACKET(media-sec-policy)#
```

5. Type inbound to enter the setting for inbound flows.

```
ACMEPACKET(media-sec-policy)# inbound
ACMEPACKET(inbound)#
```

6. mode—Enter the mode that you want to use for inbound flows. You can choose from rtp, srtp, and any. Refer to the For Incoming Flows (1082) section Mode Settings in the Media Security Policy (1082) description for details about each value.
7. protocol—Change this value to none. Use the done command to save your work, and exit the inbound configuration.
8. Type outbound to enter the setting for inbound flows.

```
ACMEPACKET(media-sec-policy)# outbound
ACMEPACKET(outbound)#
```

9. mode—Enter the mode that you want to use for outbound flows. You can choose from rtp, srtp, and any. Refer to the For Outgoing Flows (1082) section Mode Settings in the Media Security Policy (1082) description for details about each value.
10. protocol—Change this value from to none. Use the done command to save your work, and exit the outbound configuration.
11. Type done and continue.

Egress Offer Format for SDES Profile Configuration

To set the egress offer format for an SDES profile configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type media-security and press Enter.

```
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)#
```

4. Type sdes-profile and press Enter. If you are editing a pre-existing configuration, you needs to select it before you can make changes.

```
ACMEPACKET(media-security)# sdes-profile
ACMEPACKET(sdes-profile)#
```

5. egress-offer-format—Choose an egress offer format for this profile to use when you set the outbound mode in the media security policy to any. You can select one of two values:
   - If the value is same-as-ingress (default), the E-SBC leaves the profile of the media lines unchanged.
   - If the value is simultaneous-best-effort, the E-SBC inspects the incoming offer SDP and:
     - Adds an RTP/SAVP media line for any media profile that has only the RTP/AVP media profile
     - Adds an RTP/AVP media line for any media profile that has only the RTP/SAVP media profile
6. Type done to save your work and continue.

   To set the egress offer format for an SDES profile configuration:
7. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

8. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

9. Type media-security and press Enter.

```
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)#
```

10. egress-offer-format—Choose an egress offer format for this profile to use when you set the outbound mode in the media security policy to any. You can select one of two values:
    - If the value is same-as-ingress (default), the E-SBC leaves the profile of the media lines unchanged.
    - If the value is simultaneous-best-effort, the E-SBC inspects the incoming offer SDP and:
      - Adds an RTP/SAVP media line for any media profile that has only the RTP/AVP media profile
      - Adds an RTP/AVP media line for any media profile that has only the RTP/SAVP media profile
11. Type done to save your work and continue.

Supporting UAs with Different SRTP Capabilities

To support UAs with different levels of SRTP capabilities, the use-ingress-session-params parameter appears in the sdes-profile configuration. The values for this parameter allow the Oracle Enterprise Session Border Controller to accept and (where applicable) mirror the UA's proposed cryptographic session parameters:

- srtp-auth—Decides whether or not authentication is performed in SRTP
- srtp-encrypt—Decides whether or not encryption is performed in SRTP
- srtcp-encrypt—Decides whether or not encryption is performed in SRTCP

Using these possible values, the Oracle Enterprise Session Border Controller accepts the corresponding incoming session parameters.

Note: For MIKEY, this parameter and its function are reserved for future use.

Receiving Offer SDP

When the Oracle Enterprise Session Border Controller receives offer SDP with applicable session parameters, it uses the same session parameters in its answer SDP (if it can support the same). This is true even if the value for that session parameter differs from the available media security profile.

Consider this example: An SDES profile is applied for incoming direction for a media security policy configured with the srtcp-encrypt value set to enabled. With the use-ingress-session-params parameter set to srtcp-encrypt for the SDES profile, the Oracle Enterprise Session Border Controller accepts the offer SDP and also sets UNENCRYPTED_SRTCP for the cryptographic attributes in its answer SDP. When the call connects, the Oracle Enterprise Session Border Controller does not encrypt or decrypt SRTCP packets. Without the SDES profile set this way, the Oracle Enterprise Session Border Controller would reject offer SDP if any of its cryptographic attributes showed UNENCRYPTED_SRTCP in its session parameters list.

Receiving Answer SDP

When the Oracle Enterprise Session Border Controller receives answer SDP with the accepted session parameter, the value for the same session parameters that the Oracle Enterprise Session Border Controller uses might or might not be the same as the incoming value. Configuration of the outbound media security profile controls the value used because the Oracle Enterprise Session Border Controller makes offer SDP, which cannot be changed, with the session parameters based on the outgoing media security profile.

Consider this example: An SDES profile is applied for incoming direction for a media security policy configured with the srtcp-encrypt value set to enabled, so the cryptographic attributes in the SDP the Oracle Enterprise Session Border Controller sends do not have the UNENCRYPTED_SRTCP session parameters. If the UNENCRYPTED_SRTCP appears in the corresponding answer SDP it receives, the Oracle Enterprise Session Border Controller accepts it if the srtcp-encrypt value appears in the use-ingress-session-params parameter. But the Oracle Enterprise Session Border Controller still performs SRTCP encryption. When the call connects, the Oracle Enterprise Session Border Controller encrypts outgoing SRTCP packets but does not decrypt incoming SRTCP packets. So if the UA (receiving the Oracle Enterprise Session Border Controller's offer SDP) does not support SRTCP decryption, it will likely reject the offer SDP.

ACLI Instructions and Examples

To set the ingress session parameters for an SDES profile configuration:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type media-security and press Enter.

```
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)#
```

4. Type sdes-profile and press Enter. If you are editing a pre-existing configuration, you needs to select it before you can make changes.

```
ACMEPACKET(media-security)# sdes-profile
ACMEPACKET(sdes-profile)#
```

5. use-ingress-session-params—Enter the list of values for which the Oracle Enterprise Session Border Controller will accept and (where applicable) mirror the UA's proposed cryptographic session parameters:

   - srtp-auth—Decides whether or not authentication is performed in SRTP
   - srtp-encrypt—Decides whether or not encryption is performed in SRTP
   - srtcp-encrypt—Decides whether or not encryption is performed in SRTCP

   ```
   ACMEPACKET(sdes-profile)# use-ingress-session-params srtp-auth srtp-
   encrypt srtcp-encrypt
   ```

6. Type done to save your work and continue.

Refining Interoperability

To refine any remaining interoperability issues, you can use the options parameter in these configurations: media-sec-policy, sdes-profile, and mikey-profile.

Commons value to configure an option are include-local-id and include-remote-id. By default, the Oracle Enterprise Session Border Controller does not include the IDi or IDr when sending the MIKEY I_MESSAGE. And it does not set the IDr in the MIKEY R_MESSAGE. Using the options provides these results:

- include-local-id—The Oracle Enterprise Session Border Controller includes the IDi in the I_MESSAGE and the IDr in the R_MESSAGE.

  When used for the outbound direction of a media security policy, the IDi is included in the I_MESSAGE the Oracle Enterprise Session Border Controller sends. The content of the IDi is the value of the Contact header found in the INVITE message.

  When configured for the mikey-profile associated with the inbound media security policy, the Oracle Enterprise Session Border Controller includes the IDr in the R_MESSAGES it sends in response to incoming I_MESSAGES. The content of the IDr is the value of the Contact header found in the 200 OK response.
- include-remote-id—The system includes the IDr in the I_MESSAGE.

  When configured for the mikey-profile associated with the outbound media security policy, the Oracle Enterprise Session Border Controller includes the IDr in I_MESSAGES it initiates. The content of the IDr is the value of the Request-URI from the INVITE message.

Refining Interoperability Configuration

You can configure these options for media-sec-policy, sdes-profile, and mikey-profile configurations. The following uses the mikey-profile to demonstrate how to enter them.

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type media-security and press Enter.

```
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)#
```

4. Type mikey-profile and press Enter. If you are editing a pre-existing configuration, you needs to select it before you can make changes.

```
ACMEPACKET(media-security)# mikey-profile
ACMEPACKET(mikey-profile)#
```

5. options——Your entry will look like this when you add both values:

```
ACMEPACKET(mikey-profile)# options include-local-id, include-remote-id
```

You can you the plus sign (+) and the minus sign (-) to add and remove values from the options list.

To remove an value, your entry would look like this:

```
ACMEPACKET(mikey-profile)# options -include-local-id
```

To add an value, your entry would look like this:

```
ACMEPACKET(mikey-profile)# options +include-local-id
```

Multi-system Selective SRTP Pass-through

Prior to Release S-CX6.3F1, Oracle Enterprise Session Border Controller provided a single Secure Real-time Transport Protocol (SRTP) operational mode, referred to as SRTP Media Proxy Mode. In this original processing mode, each Oracle Enterprise Session Border Controller in the SRTP media path served as a proxy for the media — always decrypting inbound traffic, and encrypting outbound traffic.

Release S-CX6.3F1 introduces support for a new, alternative processing mode, referred to as Multi-system Selective SRTP Pass-through Mode. In this new mode encryption and decryption of SRTP media is handled by the SRTP endpoints, the calling and called parties. Off-load of the processor-intensive encryption/decryption provides the Oracle Enterprise Session Border Controller with the ability to handle a larger number of simultaneous SRTP sessions.

With Multi-system Selective SRTP Pass-through enabled, the Oracle Enterprise Session Border Controller can configured to selectively allow hair-pinned (spiral) SRTP packets to pass through the Oracle Enterprise Session Border Controller without encryption or decryption.

Note: hair-pinned calls are those calls where the calling and called parties are within the same realm and/or within the same sub-network.

License Requirements

Multi-system Selective SRTP Pass-through requires only the SIP license.

Hardware Requirements

On the NN4500 platform, Multi-system Selective SRTP Pass-through requires an SSM2 (for SIP/TLS signalling), and either an IPsec/SRTP NIU, or an ETC NIU.

Constraints

Multi-system Selective SRTP Pass-through support has the following constraints:

1. The realm, or realms in which the calling and called parties are located are configured for Multi-system Selective SRTP Pass-through support.
2. The call session does not require SIP—INFO to RFC 2833 tone translation.
3. The call session does not require SDES/MIKEY internetworking. Both SDES and MIKEY can be used for the exchange of SRTP keying material. Both the calling and called parties must support the same key exchange protocol.
4. Multi-system Selective SRTP Pass-through support should not be enabled in topologies where core-side application servers may change the c-line to inject a media server or some other media device in the media path. In such cases, SRTP should be terminated at the SD for each endpoint, so that the media server receives unencrypted media. In the other case, where the c-line is not subject to modification, Multi-system Selective SRTP Pass-through can be enabled.

If any of these conditions are not met, Multi-system Selective SRTP Pass-through processing cannot be provided, and the call is serviced as specified by SRTP Media Proxy Mode.

Operational Overview

To set up Multi-system Selective SRTP Pass-through, the ingress and egress Oracle Enterprise Session Border Controllers (which can, in fact, be a single Oracle Enterprise Session Border Controller) exchange the SDES or MIKEY keying material that they receive from their respective endpoint so that the Oracle Enterprise Session Border Controller peer can pass the material to its adjacent endpoint. The endpoint to endpoint exchange of keying material enables the endpoints themselves to generate encryption/decryption keys.

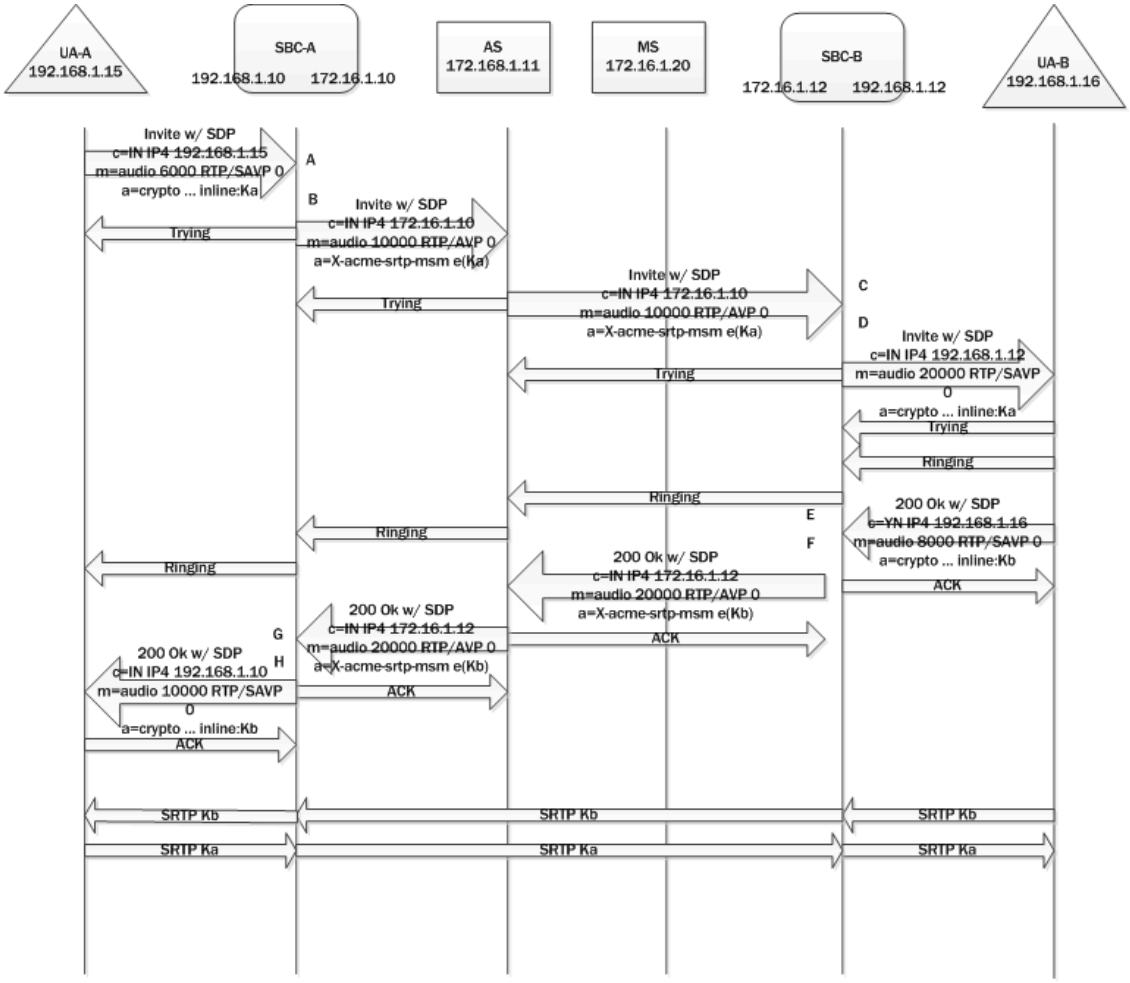
The actual exchange of keying material takes place in SIP messages (specifically, INVITE, 200 OK, and ACK) that carry offer or answer SDPs. Encrypted keying material is conveyed within a media attribute for each SRTP session. The name of the media attribute is configurable.

When either Oracle Enterprise Session Border Controller receives the encrypted keying material sent by its remote peer, it decrypts the media attribute and passes the plaintext attribute to its endpoint. Consequently, subsequent SRTP packets from the endpoints pass through the Oracle Enterprise Session Border Controllers without being decrypted and encrypted because both endpoints (now is possession of each others keying material) are able to decrypt the SRTP packets received from the other.

Call Flows

The following three sections describe call signalling for common scenarios.

Call Setup



192.168.1.15

A: The calling party sends an INVITE with SDP to SBC-A. The offer SDP contains an SDES crypto attribute within the SRTP media line.

B: Since Multi-system Selective SRTP Pass-through is enabled within the ingress realm, SD-A adds an a=X-acme-srtp-msm media attribute. The a=X-acme-srtp-msm attribute contains a cookie that includes an encryption of the SDES crypto attribute present in the SDP. The encryption is done using the shared secret configured for encrypting SRTP Pass-through information.

C: SBC-B receives the offer SDP that has the cookie sent by SBC-A. It is assumed that the proxies that forward the offer SDP sent by SBC-A preserve and forward the cookie added by SBC-A.

D: SBC-B checks if the egress realm has Multi-system Selective SRTP Pass-through enabled. If so, SBC-B decrypts the cookie using the shared secret to retrieve the SDES crypto attribute. SBC-B adds the SDES crypto attribute retrieved from the cookie to the offer SDP sent to UA-B.

E: The called party sends an answer SDP with a SDES crypto attribute on the SRTP media line to SBC-B.

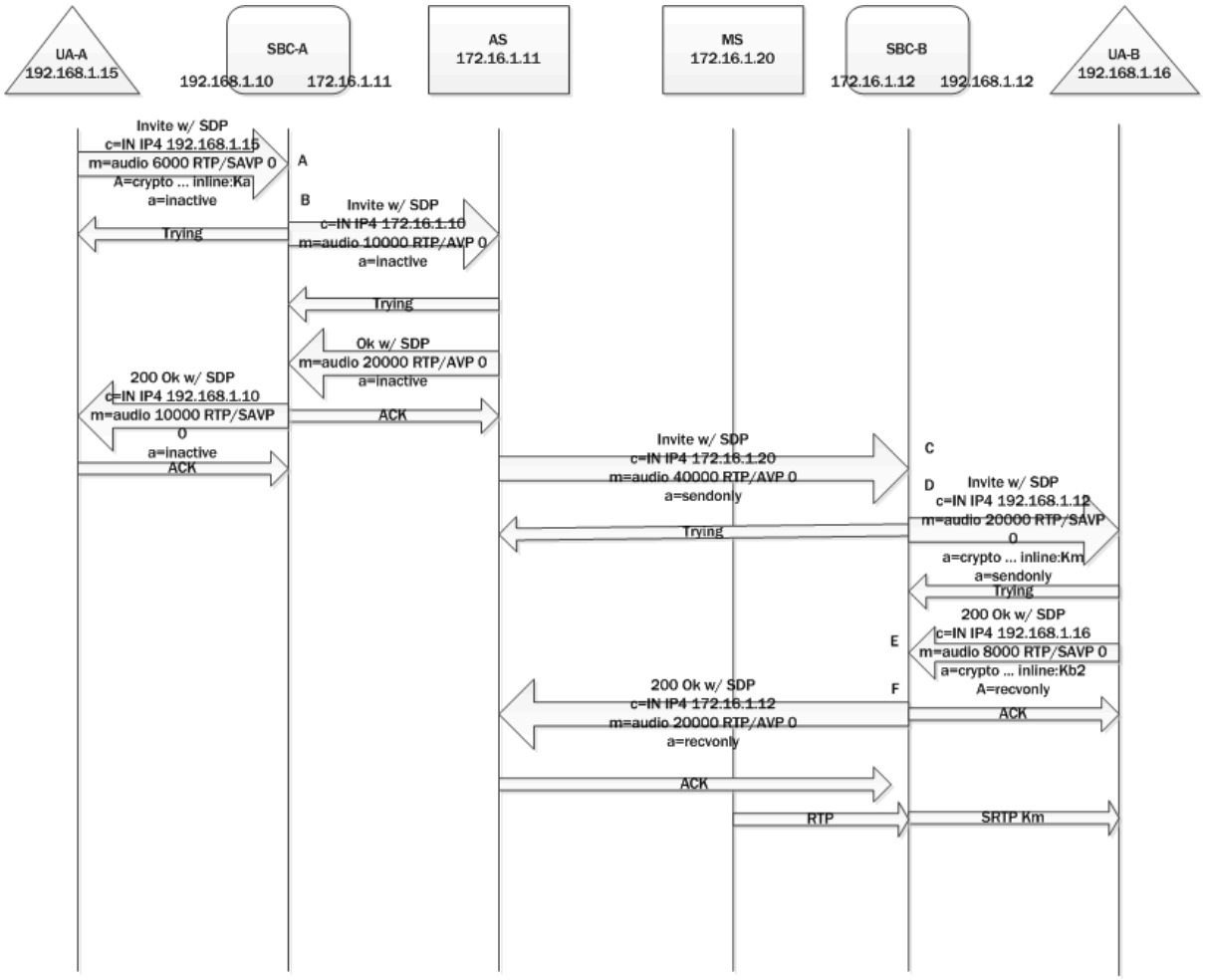
F: SBC-B checks if it has received a cookie in the offer SDP and adds the cookie to the answer SDP. The cookie contains an encryption of the SDES crypto attribute received in the answer SDP from UA-B. SBC-B does not install any SA or media policy so that SRTP packets from/to UA-B can pass through SBC-B without any decryption/ encryption.

G: SBC-A receives the answer SDP that has the cookie sent by SBC-B.

H: SBC-A decrypts the cookie using the shared secret to retrieve the SDES crypto attribute. SBC-A adds the SDES crypto attribute retrieved from the cookie to the answer SDP. SBC-A does not install any SA or media policy so that SRTP packets from/to UA-A can pass through SBC-A without any decryption/encryption.

Music on Hold

After a call is established, one of the endpoints can put the other endpoint on hold. An application server might intercept the re-INVITE from one endpoint (for putting the other on hold) and implement MoH as follows.



A: Endpoint UA-A sends an offer SDP for hold to SBC-A.

B: SBC-A forwards offer SDP without any cookie since there will be no media going from/to UA-A.

C: SBC-B receives an offer SDP that is addressed to the MS without any cookie.

D: Because there is no cookie in the ingress offer SDP, SD-B generates its own SDES crypto attributes for the egress offer SDP sent to UA-B.

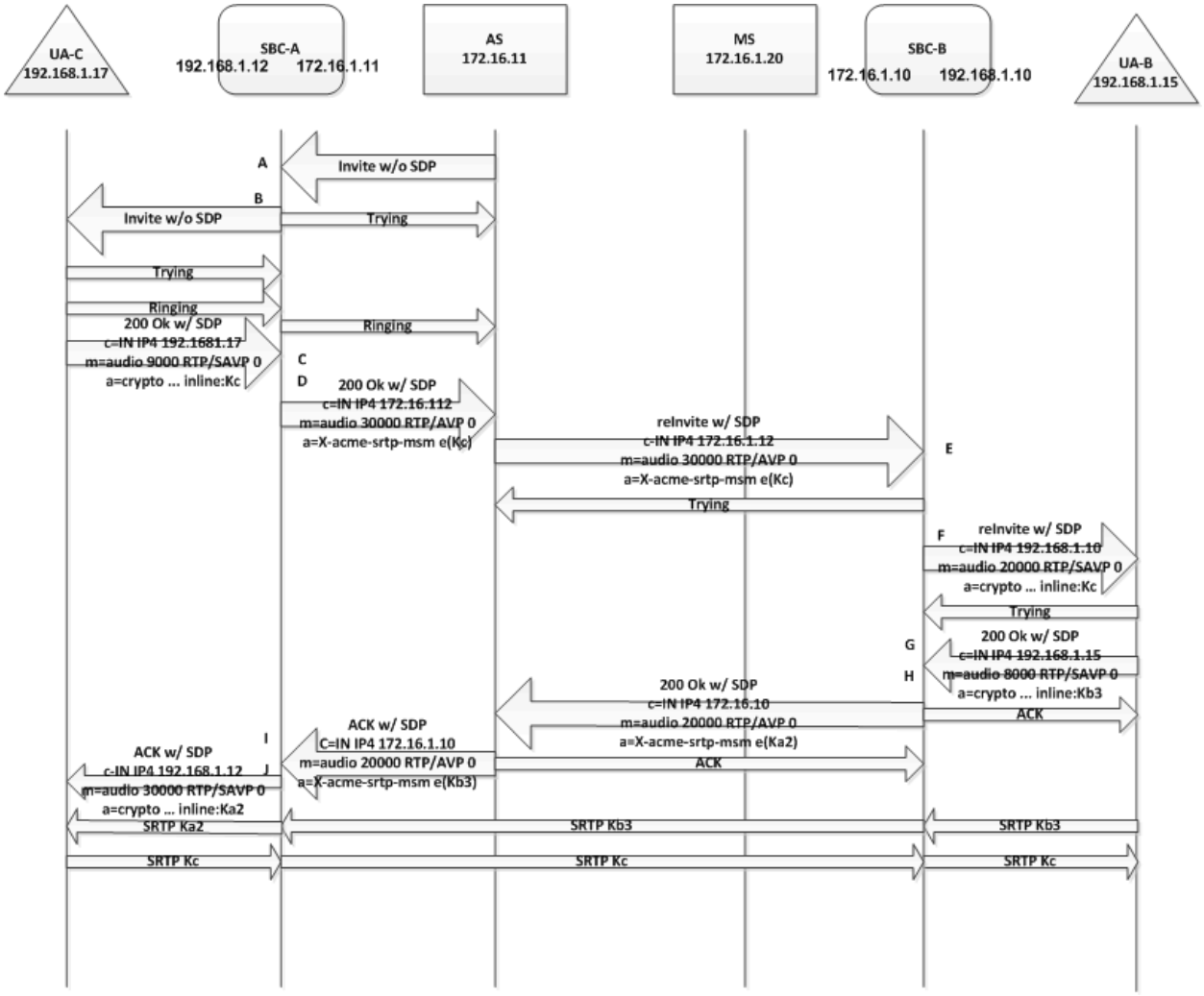
E: SBC-B receives am answer SDP from UA-B.

F: SBC-B sends its answer SDP without any cookie and encrypts SRTP packets going to UA-B. Note that there will be no SRTP packets going from UA-B to SBC-B.

As a result, MoH is heard by UA-B as it decrypts SRTP packets encrypted by SBC-B. When UA-A resumes, the steps to establish media between UA-A and UA-B will be the same as the steps used for call setup.

Once the media is reestablished between UA-A and UA-B media travelling between the two UAs will not be decrypted by either SBC.

Call Transfer



A call transfer can also happen after a call is established. For example, endpoint UA-B can transfer UA-A to another endpoint, UA-C. UA-B can make a blind transfer or an attended transfer. The following diagram describes a blind call transfer with SRTP Pass-through enabled. Note it does not show the SIP message exchange between UA-B and the Application Server to facilitate the call transfer (i.e. the INVITE from UA-B to put the call on hold and the REFER/ NOTIFY message exchange between UA-B and the Application Server). After UA-A is put on hold and the transfer target (that is, UA-C) is received from UA-B, the Application Server attempts to establish a call to UA-C. After the call to UA-C is established, the Application Server takes UA-A off hold and connects its media session to that of UA-C.

A: SBC-B receives an INVITE (from the Application Server to invite UA-C) without an offer SDP.

B: SBC-B sends an INVITE without an offer SDP to UA-C.

C: SBC-B receives a 200 OK response with an offer SDP that has a SDES crypto attribute on the SRTP media line from UA-C.

D: Since Multi-system Selective SRTP Pass-through is enabled within the ingress realm, SBC-B adds a cookie to the egress offer SDP that is sent to the core realm.

E: SBC-A receives a re-INVITE to UA-A with the offer SDP that has the cookie sent by SBC-B.

F: Since Multi-system Selective SRTP Pass-through is enabled within the ingress realm, SBC-A adds the SDES crypto attribute retrieved from the cookie to the offer SDP sent to UA-A.

G: SBC-A receives an answer SDP that has a SDES crypto attribute on the SRTP media line from UA-A.

H: SBC-A checks if it has a cookie in the offer SDP and adds the cookie to the answer SDP that is sent to the core realm. The cookie contains an encryption of the SDES crypto attribute received in the answer SDP from UA-B. SBC-A stops the encryption of any SRTP packets going to UA-B (set up for MoH) so that SRTP packets from/to UA-B can now pass through SBC-A without any decryption/encryption.

I: SBC-B receives the answer SDP that has the cookie sent by SBC-A.

H: SBC-B decrypts the cookie using the shared secret to retrieve the SDES crypto attribute. SBC-B adds the SDES crypto attribute retrieved from the cookie to the answer SDP.

After the call to UA-C is established and the call to UA-A is resumed (with the resulted media going between UA-A and UA-C) the Application Server disconnects the call with UA-A. Note that steps C-J are essentially the same steps as steps A-H in the Call Setup example. The difference is that the offer SDP from C comes to SD-B in a 200 OK response and the answer SDP sent by SBC-B to C is in the ACK.

Early Media

Different application servers may implement early media in different ways. In general, the SBC supports early media in the following way.

If the SBC receives an SDP without any cookie in a provisional response, the SBC generates its own SRTP crypto context and exchanges it with the caller via the SDP included in the provisional response. The SBC continues to decrypt and encrypt early media packets going to and from the caller. The SBC stops decrypting and encrypting only if it receives a final response with an answer SDP that signals that SRTP Pass-through should be enabled.

Multi-system Selective SRTP Pass-through with Media Release

When SRTP Pass-through is allowed, the hair-pinned media can also be released to the network if media release is configured — that is, if realm-config parameters msm-release is enabled, mm-in-realm is disabled, or mm-in-network is disabled. If SRTP Pass-through is not allowed, media release will not be allowed either.

Multi-system Selective SRTP Pass-through Configuration

Use the following procedure to enable Multi-system Selective SRTP Pass-through within a specific realm.

1. Use the following command sequence to move to realm-config Configuration Mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

2. Use the srtp-msm-passthrough parameter to enable Multi-system Selective SRTP Pass-through within a specific realm.

   By default, pass-through support is disabled.

```
ACMEPACKET(realm-config)# srtp-msm-passthrough enabled
ACMEPACKET(realm-config)#
```

3. Use done, exit, and verify-config to complete enabling Multi-system Selective SRTP Pass-through within the current realm.

   verify-config checks that the srtp-msm-password parameter has been configured, and outputs an error if it has not been configured. verify-config also checks other configuration settings that conflict with Multi-system SRTP Pass-through operation. Among these possible mis-configurations are the following.

   rfc2833-mode set to preferred on a SIP interface within a realm that has srtp-msm-passthrough enabled

   rfc2833-mode set to preferred and app-protocol set to SIP on a session-agent within a realm that has srtp-msm-passthrough enabled.

4. If required, repeat Steps 1 through 3 to enable Multi-system Selective SRTP Pass-through on additional realms.

   Use the following procedure to specify values need to support the exchange of SDES or MIKEY keying information.

5. Use the following command sequence to move to security Configuration Mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

6. Use the srtp-msm-attr-name parameter to specify the name of the media attribute used to convey SDES or MIKEY keying information within a SDP media description.

   A valid attribute name must consist of characters from the US-ASCII subset of ISO-10646/UTF-8 as specified in RFC 2327, *SDP: Session Description Protocol*. IANA-registered names should not be used. Values should begin with an X-1 prefix to prevent collision with registered values.

   In the absence of a specified attribute name, the SD provides a default value of X-acme-srtp-msm.

```
ACMEPACKET(security)# srtp-msm-attr-name X-key-material
ACMEPACKET(security)#
```

7. Use the srtp-msm-password parameter to provide the shared secret used to derive the key for encrypting SDES or MIKEY keying material that is placed in the media attribute of an SDP media description. Ingress keying material is encrypted using this shared secret before being forwarded to the network core. On egress, the encrypted keying material is decrypted with this same key.

   Allowable values are characters strings that contain a minimum of 8 and a maximum of 16 characters.

```
ACMEPACKET(security)# srtp-msm-password IsHeEleemosynary
ACMEPACKET(security)#
```

8. Use done, exit, and verify-config to complete necessary configuration.

   verify-config checks that the srtp-msm-password parameter has been configured, and outputs an error if it has not been configured. verify-config also checks other configuration settings that conflict with Multi-system SRTP Pass-through operation. Among these possible mis-configurations are the following.

   rfc2833-mode set to preferred on a SIP interface within a realm that has srtp-msm-passthrough enabled

   rfc2833-mode set to preferred and app-protocol set to SIP on a session-agent within a realm that has srtp-msm-passthrough enabled.

Statistics

The number of media sessions set up with Multi-systems Selective SRTP Pass-through is tracked and included in the output of the show mbcd statistics command.

IPSec Support

The Oracle Enterprise Session Border Controller offers IPSec for securing signaling, media, and management traffic at the network layer.

Supported Protocols

The Oracle Enterprise Session Border Controller's IPSec implementation supports all required tools for securing Internet communication via the IPSec protocol suite. The following paragraphs list and explain the protocols within the IPSec suite that the Oracle Enterprise Session Border Controller supports. This chapter does not explain how to design and choose the best protocols for your application.

AH vs. ESP

The Oracle Enterprise Session Border Controller supports the two encapsulations that IPSec uses to secure packet flows. Authentication Header (AH) is used to authenticate and validate IP packets. Authentication means that the packet was sent by the source who is assumed to have sent it.

Note: AH is incompatible with NAT. Validation means that the recipient is assured that the packet has arrived containing the original, unaltered data as sent.

ESP (Encapsulating Security Payload) provides AH's authentication and validations and extends the feature set by ensuring that the IP packet's contents remain confidential as they travel across the network. Using an encryption algorithm that both peers agree upon, ESP encrypts a full IP packet so that if intercepted, an unauthorized party cannot read the IPSec packet's contents.

Tunnel Mode vs. Transport Mode

In addition to its security encapsulations, the IPSec suite supports two modes: tunnel mode and transport mode. Tunnel mode is used most often for connections between gateways, or between a host and a gateway. Tunnel mode creates a VPN-like path between the two gateways and encapsulates the entire original IP packet. Transport mode is used to protect end-to-end communications between two hosts providing a secured IP connection and encrypts just the original payload.

Note: Traffic sent through the inner IPSec tunnel must be on the same VLAN-slot-port network-interface combination as where the outer tunnel is configured. This is because IPSec tunnel mode does not carry any L2 information for the inner packet. Once the SBC decrypts (de-tunnel) the received packet, it uses the L2 header from the original packet for the lookup. Therefore, if the SBC uses different vlan/slot/port for the inner network, lookups will fail.

Cryptographic Algorithms

IPSec works by using a symmetric key for validation and encryption. Symmetric key algorithms use the same shared secret key for encoding and decoding data on both sides of the IPSec flow. The Oracle Enterprise Session Border Controller's IPSec feature supports the following encryption algorithms:

- DES
- 3DES
- AES128CBC
- AES256CBC
- AES128CTR
- AES256CTR

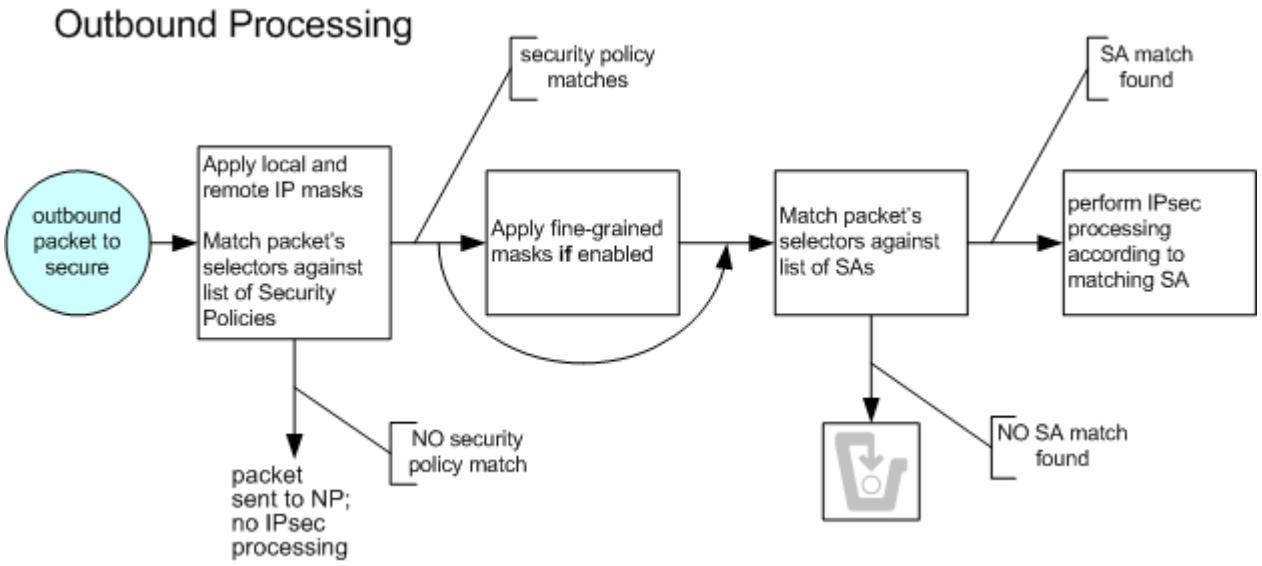
The Oracle Enterprise Session Border Controller can quickly generate keys for all of the above mentioned algorithms from the CLI. It can additionally support HMAC-SHA1 or HMAC-MD5 keyed-hash message authentication codes. Only manual keying is currently supported for both hash authentication and data encryption. Therefore, all keys must be provisioned on the Oracle Enterprise Session Border Controller by hand.

IPSec Implementation

The Oracle Enterprise Session Border Controller uses separate logic for processing IPSec packets based on whether the traffic is inbound or outbound. The configuration is divided into two pieces, the security policy and the security association (SA). Both the SA and security policies have a directional attribute which indicates if they can be used and/or reused for inbound and outbound traffic.

Outbound Packet Processing

The following diagrams show the steps the Oracle Enterprise Session Border Controller follows when processing outbound IPSec traffic. Details of each step are described in the following sections.



Security Policy

The Oracle Enterprise Session Border Controller first performs a policy lookup on outbound traffic to test if it should be subjected to IPSec rules. A security policy, local policy applicable for IPSec functionality, defines the matching criteria for outgoing network traffic to secure. It is configured on a network interface basis.

Configuring a security policy is similar to a local policy, with additional IPSec-specific parameters. Unlike a local policy, used for routing, a security policy is used for packet treatment. As with a local policy, a set of selector values is matched against the outbound flow's following characteristics:

- VLAN
- Source IP address (plus mask)
- Source IP port
- Destination IP address (plus mask)
- Destination IP port
- Transport Protocol

Each of these selection criteria can be defined by a wildcard except for the VLAN ID, which can be ignored. This flexibility aids in creating selection criteria that ranges from highly restrictive to completely permissive. In addition to the main traffic matching criteria, a priority parameter is used to prioritize the order that configured security policies are checked against. The #0 policy is checked first, #1 policy is checked next, continuing to the lowest prioritized policy being checked last.

Once the outbound traffic matches a policy, the Oracle Enterprise Session Border Controller proceeds to the next step of outbound IPSec processing. If no matching security policy is found, the default pass-through policy allows the packet to be forwarded to the network without any security processing.

Fine-grained policy Selection

After a positive match between outbound traffic and the configured selectors in the security policy, the Oracle Enterprise Session Border Controller can perform a calculation between a set of fine-grained packet selectors and the outbound packet. The fine-grained policy masking criteria are:

- Source IP subnet mask
- Destination IP subnet mask
- VLAN mask

By default, the fine-grained security policy is set to match and pass all traffic untouched to the security association (SA) portion of IPSec processing.

Fine-grained policy selection works by performing a logical AND between outbound traffic's fine-grained selectors and the traffic's corresponding attributes. The result is then used to find the matching SA. Applying a fine-grained mask has the effect of forcing a contiguous block of IP addresses and/or ports to appear as one address and or port. During the next step of IPSec processing, when an SA is chosen, the Oracle Enterprise Session Border Controller in effect uses one SA lookup for a series of addresses. Without fine-grained policy selection, unique SAs must always be configured for outbound packets with unique security policy selectors.

Security Associations

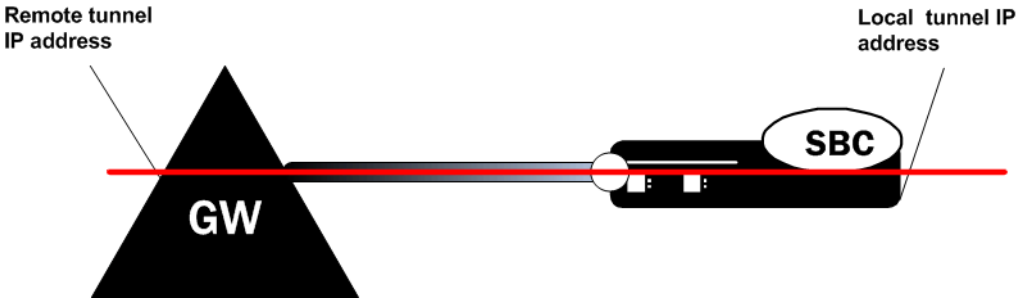
After the Oracle Enterprise Session Border Controller determines that outgoing traffic is subject to IPSec processing, and optionally applies fine-grained masking, an SA lookup is performed on the traffic. An SA is the set of rules that define the association between two endpoints or entities that create the secured communication. To choose an SA, the Oracle Enterprise Session Border Controller searches for a match against the outgoing traffic's SA selectors. SA selectors are as follows:

- VLAN
- Source IP address
- Source IP port
- Destination IP address
- Destination IP port
- Transport Protocol

If there is a match, the Oracle Enterprise Session Border Controller secures the flow according to security parameters defined in the SA that the Oracle Enterprise Session Border Controller chooses. The packet is then forwarded out of the Oracle Enterprise Session Border Controller. If no match is found, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG.

Secure Connection Details

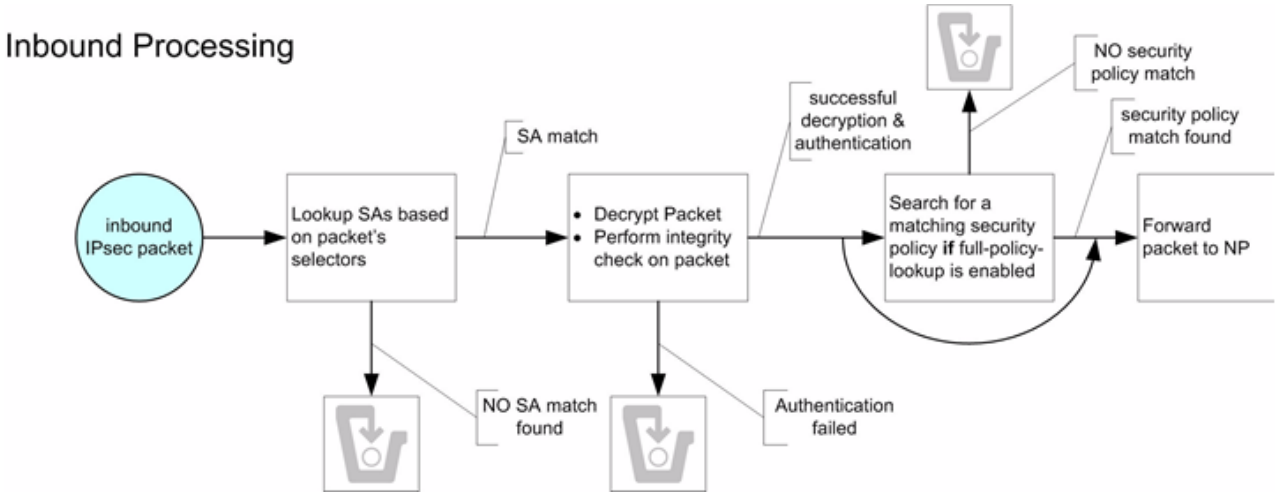
Several parameters define an IPSec connection between the Oracle Enterprise Session Border Controller and a peer. When planning an IPSec deployment, the primary architectural decisions are which IPSec protocol and mode to use. The two choices for IPSec protocol are ESP or AH, and the two choices for IPSec mode are either tunnel or transport. IPSec protocol and mode are both required for an SA configuration. When creating an IPSec tunnel (tunnel mode), the SA must also define the two outside IP addresses of the tunnel.



The authentication algorithm and the authentication key must always be configured. The Oracle Enterprise Session Border Controller supports hmac-md5 or hmac-sha1 authentication algorithms. Because only manual keying is supported, the key must be entered by hand. When encryption is required, the encryption algorithm and the encryption key must be configured. The Oracle Enterprise Session Border Controller supports des, 3des, aes-128-cbc, aes-256-cbc, aes-128-ctr, and aes-256-ctr encryption algorithms. When using the two encryption protocols that operate in AES counter mode (RFC 3686), an additional nonce value is required. In addition, the security parameter index (SPI) must be configured for each SA. All SPI values mst be unique as well, across all SAs.

Inbound Packet Processing

The following diagram shows the steps the system follows when processing inbound IPSec traffic. Details of each step are described in the following sections.



IP Header Inspection

Processing inbound IPSec packets begins by the Oracle Enterprise Session Border Controller inspecting an inbound IP packet's headers. If the packet is identified as IPSec traffic, as determined by the presence of an AH or ESP header, an SA policy lookup is performed. If the traffic is identified as non-IPSec traffic, as determined by the lack of an IPSec-type (AH or ESP) header, it still is subject to a policy lookup. However, due to the default allow policy, the packet is forwarded directly to the NP, without any security processing.

SA Matching

The Oracle Enterprise Session Border Controller proceeds to match the inbound IPSec traffic's selectors against configured SAs. Inbound selector masking is performed where noted. These selectors are:

- VLAN (plus mask)
- Source IP address (plus mask)
- Source IP port
- Destination IP address (plus mask)
- Destination IP port
- Transport Protocol
- SPI

If no matching SA is found, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG. When the Oracle Enterprise Session Border Controller finds a matching SA, the packet is authenticated and decrypted according to the configuration and sent to the Oracle Enterprise Session Border Controller's NP for continued processing.

Inbound Full Policy Lookup

Inbound traffic can optionally be subjected to a full policy lookup, after decryption and authentication. A full policy lookup checks if a security policy exists for this inbound traffic before the Oracle Enterprise Session Border Controller sends the decrypted packet to the NP. If no matching security policy is found, even after a successful SA match, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG.

Full policy lookups are useful for traffic filtering. If you wish to drop traffic not sent to a specific port (e.g. drop any traffic not sent to port 5060), a security policy with specific remote-port-match parameter would be used to define what is valid (i.e., not dropped).

HA Considerations

Anti-replay mechanisms, running on IPSec peers, can cause instability with the Oracle Enterprise Session Border Controllers configured in an HA pair. The anti-replay mechanism ensures that traffic with inconsistent (non-incrementing) sequence numbers is labeled as insecure, assuming it could be part of a replay attack. Under normal circumstances, this signature causes the remote endpoint to drop IPSec traffic.

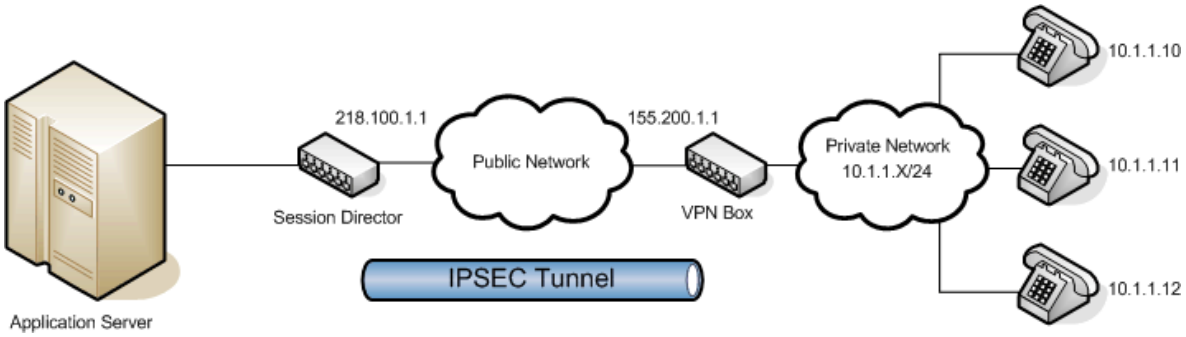
When a failover occurs between HA peers, the newly-active system starts sending traffic with the IPSec sequence number starting at 0. A remote system's anti-replay mechanism observes this and labels the traffic as insecure. It is therefore recommended that anti-replay protection not be used with Oracle Enterprise Session Border Controllers in an HA configuration. This situation does not create any problems as long as IPSec peers are not configured to use anti-replay mechanisms.

Packet Size Considerations

The security processor supports receipt of jumbo frames up to 9K (9022 bytes with VLANs, 9018 without). Under normal operation the default outgoing maximum packet size of 1500 bytes is used. This packet size includes the IPSec headers, which will result in less space for packet data (SIP signaling, RTP, etc...).

IPSec Application Example

In this example, the Oracle Enterprise Session Border Controller terminates an IPSec tunnel. The remote side of the tunnels is a dedicated VPN appliance in the public Internet. Behind that VPN appliance are three non-IPSec VoIP phones. In this scenario, the VPN box maintains the IPSec tunnel through which the phones communicate with the Oracle Enterprise Session Border Controller.



Without the fine-grained option (or alternatively IKE), an SA entry would need to be configured for each of the three phones, communicating over the IPSec tunnel (resulting in 3 tunnels).

This does not scale for manual-keying with a large number of endpoints. Using the fine-grained configuration as well as the inbound SA mask allows any number of endpoints on the 10.1.1.X network to use a single security association (a coarse-grain configuration). The configuration in this example follows:

A packet sent from the Oracle Enterprise Session Border Controller to any of the phones will match the policy pol1. The remote-ip-mask parameter of the fine-grained configuration will then be masked against the remote-ip, resulting in a SA selector value of 10.1.1.0. This matches security-association sa1, and the packet will be secured and sent over the tunnel. The tunnel-mode addresses in the security-association represent the external, public addresses of the termination points for the IPSec tunnel.

Packets returning from the 10.1.1.0 side of the IPSec tunnel will first match the tunnel-mode local-ip-addr of 218.100.1.1. The packets will then be decrypted using the SA parameters, and the tunneled packet will be checked against the remote-ip-addr field of the SA.

If the fine-grained mask had not been used, three discrete SAs would have to be configured: one for each of the three phones.

```
ACMEPACKET(manual)# show
        manual
                name                            assoc1
                spi                             1516
                network-interface               lefty:0
                local-ip-addr                   100.20.50.7
                remote-ip-addr                  100.25.56.10
                local-port                      60035
                remote-port                     26555
                trans-protocol                  ALL
                ipsec-protocol                  esp
```

```
                direction                       both
                ipsec-mode                      tunnel
                auth-algo                       hmac-md5
                encr-algo                       des
                auth-key
                encr-key
                aes-ctr-nonce                   0
                tunnel-mode
                        local-ip-addr                   100.20.55.1
                        remote-ip-addr                  101.22.54.3
                last-modified-date              2007-04-30 16:04:46
```

IPSec Configuration

The following example explains how to configure IPSec on your Oracle Enterprise Session Border Controller.

Note: If you change the phy-interface slot and port associated with any SAs or SPDs, the Oracle Enterprise Session Border Controller must be rebooted for the changes to take effect.

Configuring an IPSec Security Policy

To configure an IPSec security policy:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter to access the security path of the configuration menu.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type ipsec and press Enter.

```
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)#
```

4. Type security-policy and press Enter. The prompt changes to let you know that you can begin configuration.

```
ACMEPACKET(ipsec)# security-policy
ACMEPACKET(security-policy)#
```

5. name—Enter a name for this security policy. This parameter is required and has no default.
6. network-interface—Enter the network interface and VLAN where this security policy applies in the form: interface-name:VLAN
7. priority—Enter the priority number of this security policy. The default value is zero (0). The valid range is:
   - Minimum—0
   - Maximum—254
8. action—Enter the action the Oracle Enterprise Session Border Controller should take when this policy matches outbound IPSec traffic. The default value is ipsec. The valid values are:
   - ipsec—Continue processing as IPSec traffic
   - allow—Forward the traffic without any security processing
   - discard—Discard the traffic
9. direction—Enter the direction of traffic this security policy can apply to. The default value is both. The valid values are:
   - in—This security policy is valid for inbound traffic
   - out—This security policy is valid for outbound traffic
   - both—This security policy is valid for inbound and outbound traffic

   To define the criteria for matching traffic selectors for this security policy:
10. local-ip-addr-match—Enter the source IP address to match. The default value is 0.0.0.0.

11. remote-ip-addr-match—Enter the destination IP address to match. The default value is 0.0.0.0.
12. local-port-match—Enter the source port to match. A value of 0 disables this selector. The default value is zero (0). The valid range is:
    - Minimum—0
    - Maximum—65535
13. remote-port-match—Enter the destination port to match. A value of 0 disables this selector. The default value is zero (0). The valid range is:
    - Minimum—0
    - Maximum—65535
14. trans-protocol-match—Enter the transport protocol to match. The default value is all. The valid values are:
    - UDP | TCP | ICMP | ALL
15. local-ip-mask—Enter the source IP address mask, in dotted-decimal notation. The default value is 255.255.255.255.
16. remote-ip-mask—Enter the remote IP address mask, in dotted-decimal notation. The default value is 255.255.255.255.
17. Save your work using the ACLI done command.

Defining Outbound Fine-Grained SA Matching Criteria

To define outbound fine-grained SA matching criteria:

1. From within the security policy configuration, type outbound-sa-fine-grained-mask and press Enter. The prompt changes to let you know that you can begin configuration.
2. local-ip-mask—Enter the fine-grained source IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
3. remote-ip-mask—Enter the fine-grained destination IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
4. local-port-mask—Enter the local port mask for this security policy. The default value for this parameter is 0. The valid range is:
    - Minimum—0
    - Maximum—65535
5. remote-port-mask—Enter the remote port mask for this security policy. The default value for this parameter is 0. The valid range is:
    - Minimum—0
    - Maximum—65535
6. trans-protocol-mask—Enter the transport protocol mask for this security policy. The default value for this parameter is 0. The valid range is:
    - Minimum—0
    - Maximum—255
7. vlan-mask—Enter the fine-grained VLAN mask to apply to outbound IP packets for SA matching. The default is 0x000 (disabled). The valid range is:
    - 0x000 - 0xFFF
8. Save your work using the ACLI done command.

Configuring an IPSec SA

To configure an IPSec SA:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type security and press Enter to access the security path of the configuration menu.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```

3. Type ipsec and press Enter.

```
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)#
```

4. Type security-association and press Enter.

```
ACMEPACKET(ipsec)# security-association
ACMEPACKET(security-association)#
```

5. Type manual and press Enter. The prompt changes to let you know that you can begin configuration.

```
ACMEPACKET(security-association)# manual
ACMEPACKET(manual)#
```

6. name—Enter a name for this security policy.
7. network-interface—Enter the network interface and VLAN where this security policy applies in the form: interface_name:VLAN
8. direction—Enter the direction of traffic this security policy can apply to. The default value is both. Valid values are:
   - in | out | both
9. Save your work using the ACLI done command.

Defining Criteria for Matching Traffic Selectors per SA

To define the criteria for matching traffic selectors for this SA:

1. From within the manual portion of the security association configuration, you need to set the parameters described in this process.

```
ACMEPACKET(security-association)# manual
ACMEPACKET(manual)#
```

2. local-ip-addr—Enter the source IP address to match.
3. remote-ip-addr—Enter the destination IP address to match.
4. local-port—Enter the source port to match. A value of 0 disables this selector. The default value is 0, disabling this parameter. The valid range is:
   - Minimum—0
   - Maximum—65535
5. remote-port—Enter the destination port to match. A value of 0 disables this selector. The default value is 0, disabling this parameter. The valid range is:
   - Minimum—0
   - Maximum—65535
6. trans-protocol—Enter the transport protocol to match for traffic selectors for this SA. The default value is ALL. The valid values are:
   - UDP | TCP | ICMP | ALL
7. ipsec-protocol—Select the IPSec protocol to use for this SA configuration. The default value for this parameter is esp. Valid values are:
   - esp | ah
8. spi—Enter the security parameter index. The default value is 256. The valid range is:
   - Minimum—256
   - Maximum—2302
9. ipsec-mode—Enter the IPSec mode of this SA. The default value is transport. The valid values are:

- tunnel | transport

10. auth-algo—Enter the IPSec authentication algorithm for this SA. The default value is null. The valid values are:
    - hmac-md5 | hmac-sha1 | null
11. auth-key—Enter the authentication key for the previously chosen authentication algorithm for this SA.

    Note: The specified auth-key value will be encrypted in the configuration and will no longer be visible in clear-text.
12. encr-algo—Enter the IPSec encryption algorithm for this SA. The default value is null. The valid values are:
    - des | 3des | aes-128-cbc | aes-256-cbc | aes-128-ctr | aes-256-ctr | null
13. encr-key—Enter the encryption key for the previously chosen encryption algorithm for this SA.

    Note: The specified encr-key value will be encrypted in the configuration and will no longer be visible in clear-text.
14. aes-ctr-nonce—Enter the AES nonce if aes-128-ctr or aes-256-ctr were chosen as your encryption algorithm. The default value is 0.

Defining Endpoints for IPSec Tunnel Mode

To define endpoints for IPSec tunnel mode:

1. From within the manual portion of the security association configuration, you need to set the parameters described in this process.

```
ACMEPACKET(security-association)# manual
ACMEPACKET(manual)#
```

2. local-ip-addr—Enter the local public IP address which terminates the IPSec tunnel.
3. remote-ip-addr—Enter the remote public IP address which terminates the IPSec tunnel.
4. Save your work using the ACLI done command.

Real-Time IPSec Process Control

The notify secured commands force the IPSec application to perform tasks in real-time, outside of the Oracle Enterprise Session Border Controller reloading and activating the running configuration. The notify secured usage is as follows:

```
notify secured [activateconfig | nolog | log | debug | nodebug]
```

The following arguments perform the listed tasks:

- nolog—Disables secured logging
- log—Enables secured logging
- debug—Sets secured log level to DEBUG
- nodebug—Sets secured log level to INFO

Key Generation

The generate-key command generates keys for the supported encryption or authentication algorithms supported by the Oracle Enterprise Session Border Controller's IPSec implementation. The generate-key commands generate random values which are not stored on the Oracle Enterprise Session Border Controller, but are only displayed on the screen. This command is a convenient function for users who would like to randomly generate encryption and authentication keys. The generate-key usage is as follows:

```
generate-key [hmac-md5 | hmac-sha1 | aes-128 | aes-256 | des | 3des]
```

IDS Reporting

The Oracle Enterprise Session Border Controller supports a wide range of intrusion detection and protection capabilities for vulnerability and attack profiles identified to date. The IDS reporting feature is useful for enterprise customers requirement to report on intrusions and suspicious behavior that it currently monitors.

IDS Licensing

This feature requires the IDS Reporting license.

Note: The following capabilities and restrictions of the license:

- The following configuration parameters located in the access control and media manager configuration elements are only visible after installing the license:
  - trap-on-demote-to-deny
  - syslog-on-demote-to-deny
  - cac-failure-threshold
  - untrust-cac-failure-threshold
- Endpoint demotions based on admission control failures are only a valid option with the IDS License.
- The presence of the IDS license makes the apSysMgmtInetAddrWithReasonDOSTrap trap available and the apSysMgmtExpDOSTrap unavailable. WIthout an IDS licence installed, only the apSysMgmtExpDOSTrap trap is available.
- The Trust->Untrust and Untrust-Deny counters in the SIP and MGCP ACLs' statistics are visible regardless of the IDS license's presence.
- The Demote Trust-Untrust and Demote Untrust-Deny collect records in the SIP and MGCP ACL HDR groups are visible regardless of the IDS license's presence.
- A GET operation can be preformed on the two MIB entries to view the global endpoint counter for Demotions from Trusted to Untrusted and from Untrusted to Deny regardless of the IDS license's presence
- On Acme Packet 3820 systems, the DOS license must be installed in addition to the IDS license in order to enable all features described in this section.

Basic Endpoint Demotion Behavior

Each session agent or endpoint is promoted or demoted among the trusted, untrusted, and denied queues depending on the trust-level parameter of the session agent or realm to which it belongs. Users can also configure access control rules to further classify signaling traffic so it can be promoted or demoted among trust queues as necessary.

An endpoint can be demoted in two cases:

1. Oracle Enterprise Session Border Controller receiving too many signaling packets within the configured time window (maximum signal threshold in realm config or access control)
2. Oracle Enterprise Session Border Controller receiving too many invalid signaling packets within the configured time window. (invalid signal threshold in realm config or access control)

Endpoint Demotion Reporting

The Oracle Enterprise Session Border Controller counts the number of endpoint or session agent promotions and demotions. Further, the Oracle Enterprise Session Border Controller counts when endpoints or session agents transition from trusted to untrusted and when endpoints transition from untrusted to denied queues. These counts are maintained for SIP and MGCP signaling applications. They appear as the Trust->Untrust and Untrust->Deny rows in the show sipd acls and show mgcp acls commands.

SNMP Reporting

These per-endpoint counters are available under APSYSMGMT-MIB -> acmepacketMgmt -> apSystemManagementModule -> apSysMgmtMIBObjects -> apSysMgmtMIBGeneralObjects.

MIB NAME	MIB OID	PURPOSE
apSysSipEndptDemTrustToUntrust	.1.3.6.1.4.1.9148.3.2.1.1.19	Global counter for SIP endpoint demotions from trusted to untrusted.
apSysSipEndptDemUntrustToDeny	.1.3.6.1.4.1.9148.3.2.1.1.20	Global counter for SIP endpoint demotions from untrusted to denied.
apSysMgcpEndptDemTrustToUntrust	.1.3.6.1.4.1.9148.3.2.1.1.21	Global counter for MGCP endpoint demotions from trusted to untrusted.
apSysMgcpEndptDemUntrustToDeny	.1.3.6.1.4.1.9148.3.2.1.1.22	Global counter for MGCP endpoint demotions from untrusted to denied.

HDR Reporting

The SIP (sip-ACL-oper) and MGCP (mgcp-oper) HDR ACL status collection groups include the following two metrics:

- Demote Trust-Untrust (Global counter of endpoint demotion from trusted to untrusted queue)
- Demote Untrust-Deny (Global counter of endpoint demotion from untrusted to denied queue)

Endpoint Demotion SNMP Traps

An SNMP trap can be sent when the Oracle Enterprise Session Border Controller demotes an endpoint to the denied queue. This is set by enabling the trap on demote to deny parameter located in the media manager config configuration element.

When the IDS license is installed and the trap on demote to deny parameter is enabled, apSysMgmtInetAddrWithReasonDOSTrap trap is sent. This trap supersedes the apSysMgmtInetAddrDOSTrap trap.

When the IDS license is installed and the trap on demote to deny parameter is disabled the apSysMgmtInetAddrWithReasonDOSTrap trap is not sent from the Oracle Enterprise Session Border Controller, even when an endpoint is demoted to the denied queue.

This apSysMgmtInetAddrWithReasonDOSTrap contains the following data:

- apSysMgmtDOSInetAddressType—Blocked IP address family (IPv4 or IPv6)
- apSysMgmtDOSInetAddress—Blocked IP address
- apSysMgmtDOSRealmID—Blocked Realm ID
- apSysMgmtDOSFromURI—The FROM header of the message that caused the block (If available)
- apSysMgmtDOSReason—The reason for demoting the endpoint to the denied queue: This field can report the following three values:
  - Too many errors
  - Too many messages
  - Too many admission control failures

Note: By default, this parameter is enabled for upgrade configurations, as the current behavior is to send a trap for every endpoint that is demoted to deny. However, for a new configuration created, the value to this configuration is disabled.

Trusted to Untrusted Reporting

Endpoints, however, transition to an intermediate state, untrusted, prior to being denied service. The Oracle Enterprise Session Border Controller provides an ACLI parameter, trap-on-demote-to-untrusted, that generates an SNMP trap when a previously trusted endpoint transitions to the untrusted state. Trap generation is disabled by default.

SNMP Reporting

Endpoint state transitions continue to be tracked by two counters available under APSYSMGMT-MIB -> acmepacketMgmt -> apSystemManagementModule -> apSysMgmtMIBObjects -> apSysMgmtMIBGeneralObjects.

MIB NAME	MIB OID	PURPOSE
apSysSipEndptDemTrustToUntrust	.1.3.6.1.4.1.9148.3.2.1.1.19	Global counter for SIP endpoint demotions from trusted to untrusted.
apSysSipEndptDemUntrustToDeny	.1.3.6.1.4.1.9148.3.2.1.1.20	Global counter for SIP endpoint demotions from untrusted to denied.

Endpoint Demotion Trusted-to-Untrusted SNMP Trap

The system can generate an SNMP trap when an endpoint transitions from the trusted to the untrusted state. The trap is structured as follows.

```
apSysMgmtInetAddrTrustedToUntrustedDOSTrap NOTIFICATION-TYPE
OBJECTS { apSysMgmtDOSInetAddressType,
apSysMgmtDOSInetAddress,
apSysMgmtDOSRealmID,
apSysMgmtDOSFromUri,
apSysMgmtDOSReason }
STATUS current
DESCRIPTION
"This trap is generated when an IP is placed on a untrusted list from trusted
list, and provides the ip address that has been demoted, the realm-id of that
IP, (if available) the URI portion of the SIP From header of the message that
caused the demotion."
::= { apSysMgmtDOSNotifications 5 }
```

The trap OID is 1.3.6.1.4.1.9148.3.2.8.0.5.

Endpoint Demotion Syslog Message

A Syslog message can be generated when an endpoint is demoted. Setting the media manager config -> syslog-on-demote-to-deny parameter to enabled writes and endpoint demotion warning to the syslog every time an endpoint is demoted to the denied queue. By default, this configuration option is set to disabled. The syslog message has a WARNING Level and looks like this:

```
Jan 15 12:22:48 172.30.60.12 ACMESYSTEM sipd[1c6e0b90] WARNING SigAddr[access:
168.192.24.40:0=low:DENY] ttl=3632 guard=798 exp=30 Demoted to Black-List
(Too many admission control failures)
```

Event Log Notification Demotion from Trusted to Untrusted

You can enable your Oracle Enterprise Session Border Controller to provide event log notification (a syslog message) any time it demotes an endpoint from trusted to untrusted. The log message contains this data: IP address of the demoted endpoint, the endpoint's configured trust level, and the reason for demotion.

To use this feature, the intrusion detection system (IDS) Reporting license must be installed and your media manager configuration must be enabled to send the log messages. This feature is enabled with the syslog-on-demote-to-untrusted parameter in the media manager.

Endpoint Demotion Configuration

To configure the Oracle Enterprise Session Border Controller to send traps and/or write syslog messages on endpoint demotion:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter to access the media-level configuration elements.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type media-manager and press Enter.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. trap-on-demote-to-deny—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to send the apSysMgmtInetAddrWithReasonDOSTrap trap when applicable.
5. syslog-on-demote-to-deny—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to write an endpoint demotion warning message to the syslog.
6. syslog-on-demote-to-untrusted—Change this parameter from disabled (default), to enabled so the Oracle Enterprise Session Border Controller will generate event notifications (syslog messages) when an endpoint becomes untrusted. For this capability to work, the IDS license must be installed on your system.
7. trap-on-demote-to-untrusted—Set this parameter to enabled for the Oracle Enterprise Session Border Controller to send the apSysMgmtInetAddrTrustedToUntrustedDOSTrap when the endpoint identified within the trap transitions from the trusted to untrusted state.
8. Save your work.

Endpoint Demotion due to CAC overage

The Oracle Enterprise Session Border Controller can demote endpoints from trusted to untrusted queues when CAC failures exceed a configured threshold. The Oracle Enterprise Session Border Controller can also demote endpoints from untrusted to denied queues when CAC failures exceed a another configured threshold.

The Oracle Enterprise Session Border Controller maintains CAC failures per-endpoint. The CAC failure counter is incremented upon certain admission control failures only if either one of cac-failure-threshold or untrust-cac-fail-threshold is non-zero.

The cac failure threshold parameter is available in the access control and realm configuration elements. Exceeding this parameter demotes an endpoint from the trusted queue to the untrusted queue. The untrust cac-failure-threshold parameter is available in the access control and realm configuration elements. Exceeding this parameter demotes an endpoint from the untrusted queue to the denied queue.

If both the cac failure threshold and untrusted cac failure threshold are configured to 0, then admission control failures are considered and counted as invalid signaling messages for determining if the invalid-signal-threshold parameter value has been exceeded.

CAC Attributes used for Endpoint Demotion

The Oracle Enterprise Session Border Controller determines CAC failures only by considering the calling endpoint’s signaling messages traversing the calling realms' configuration. If an endpoint exceeds the following CAC thresholds, the Oracle Enterprise Session Border Controller will demote the endpoint when the CAC failure thresholds are enabled.

1. sip-interface user CAC sessions (realm-config > user-cac-sessions)
2. sip-interface user CAC bandwidth (realm-config > user-cac-bandwidth)
3. External policy server rejects a session

Authentication Failures used for Endpoint Demotion

If an endpoint fails to authenticate with the Oracle Enterprise Session Border Controller using SIP HTTP digest authentication OR endpoint fails authentication with an INVITE with authentication incase registration-caching is disabled, and receives back a 401 or 407 response from the registrar

When the Oracle Enterprise Session Border Controller receives a 401 or 407 message from the registrar in response to one of the following conditions, the endpoint attempting authentication is demoted.

- endpoint fails to authenticate with the Oracle Enterprise Session Border Controller using SIP HTTP digest authentication
- endpoint fails to authenticate with the Oracle Enterprise Session Border Controller using INVITE message when registration-caching is disabled

Endpoint Demotion Configuration on CAC Failures

To configure endpoint demotion on CAC failures:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type access-control and press Enter.

```
ACMEPACKET(session-router)# access-control
ACMEPACKET(access-control)#
```

   If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. cac-failure-threshold—Enter the number of CAC failures for any single endpoint that will demote it from the trusted queue to the untrusted queue.
5. untrust-cac-failure-threshold—Enter the number of CAC failures for any single endpoint that will demote it from the untrusted queue to the denied queue.
6. Save your work.

IDS Phase 2 (Advanced Reporting)

This feature supplements the IDS reporting and protection services. IDS Phase 2 provides enterprise users with additional tools to identify, monitor, and control suspicious, and possibly, malicious traffic patterns. IDS Phase 2 requires the IDS Advanced Reporting license.

License Requirements

IDS Phase 2 requires the IDS Advanced Reporting license, which is different from the IDS license introduced in the Net-Net S-C6.2.0 release. Access to new services described in this Technical Notice requires that the original IDS license be upgraded to the IDS Advanced Reporting license.

Rejected SIP Calls

IDS Phase 2 provides tools to monitor and record rejected SIP calls. A sudden or gradual increase in such calls can, but need not, indicate malicious intent.

IDS Phase 2 provides a global counter that increments with each SIP INVITE or REGISTER message that is rejected by the Acme Packet Oracle Enterprise Session Border Controller, and offers the option of generating a syslog message in response to call rejection.

Rejected Calls Counter

The rejected calls counter is a 32-bit global counter that records the total number of rejected SIP calls. Such calls have been rejected by the Oracle Enterprise Session Border Controller with the following response codes: 400, 403, 404, 405, 408, 413, 416, 417, 420, 423, 480, 481, 483, 484, 485, 488, 494, 500, 503, 505, and 604. These response codes may change with future software revisions.

The current value of the rejected calls counter is accessible via SNMP, Historical Data Recording (HDR), or the ACLI.

apSysMgmtGeneralObjects Table (1.3.6.1.4.1.9148.3.2.1.1)		
Object Name	**Object OID**	**Description**
apSysSipTotalCallsRejected	1.3.6.1.4.1.9148.3.2.1.1.25	Global counter for SIP calls that are rejected by the SBC

The sip-error HDR collection group contains a new reporting field, Call Rejects, which contains the value of the global rejected calls counter.

The ACLI command show sipd errors displays the contents of the rejected calls counter.

```
ACMEPACKET# show sipd errors
12:29:13-131
SIP Errors/Events             ---- Lifetime ----
                        Recent      Total  PerMax
SDP Offer Errors             0          0       0
SDP Answer Errors            0          0       0
Drop Media Errors            0          0       0
Transaction Errors           0          0       0
Application Errors           0          0       0
Media Exp Events             0          0       0
Early Media Exps             0          0       0
Exp Media Drops              0          0       0
Expired Sessions             0          0       0
Multiple OK Drops            0          0       0
Multiple OK Terms            0          0       0
Media Failure Drops          0          0       0
Non-ACK 2xx Drops            0          0       0
Invalid Requests             0          5       2
Invalid Responses            0          0       0
Invalid Messages             0          0       0
CAC Session Drop             0          0       0
Nsep User Exceeded           0          0       0
Nsep SA   Exceeded           0          0       0
CAC BW Drop                  0          0       0
Calls Rejected               0          0       0 <--
```

Syslog Reporting of Rejected Calls

Users can choose to send a syslog message in response to the rejection of a SIP call. In the default state, rejected calls are not reported to syslog.

Use the following ACLI command sequence to enable syslog reporting of rejected SIP calls.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)# syslog-on-call-reject enable
```

The syslog-on-call-reject attribute, which is disabled by default, enables the generation of a syslog message in response to the rejection of a SIP call.

Use done, exit, and verify-config to complete this configuration.

Syslog messages issued in response to call rejection contain the following call-related information.

- SIP status code indicating rejection cause
- SIP method name (INVITE or REGISTER)
- Reason for denial
- Realm of calling endpoint
- Applicable local response map
- Content of Reason header (if present)
- From URI of calling endpoint

- Target URI of called endpoint
- Source and Destination IP address and port
- Transport type

The following are sample syslog messages issued in response to call rejections.

Dec 8 06:05:42 172.30.70.119 deimos sipd[205bfee4] ERROR [IDS_LOG]INVITE from source 172.16.18.100:5060 to dest 172.16.101.13:5060[UDP] realm=net172; From=sipp <sip:sipp@172.16.18.100:5060>;tag=13890SIPpTag001; target=sip:service@172.16.101.13:5060 rejected!; status=483 (Too Many Hops)

Dec 10 15:09:28 172.30.70.119 deimos sipd[2065ace8] ERROR [IDS_LOG]INVITE from source 172.16.18.5:5060 to dest 172.16.101.13:5060[UDP] realm=net172; From=sipp <sip:sipp@172.16.18.5:5060>;tag=10015SIPpTag001; target=sip:service@172.16.101.13:5060 rejected!; status=488 (sdp-address-mismatch); error=sdp address mismatch

IDS syslog messages that report rejected calls and those that report endpoint demotions now contain a string IDS_LOG, to facilitate their identification as IDS-related messages. With IDS Phase 2, IDS messages reporting either endpoint demotions or call rejections can be sent to specific, previously-configured syslog servers.

In topologies that include multiple syslog servers, use the following procedure to enable delivery of IDS-related messages to one or more specific syslog servers.

1. Use the following command sequence to move to syslog-config Configuration Mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# syslog-servers
ACMEPACKET(syslog-config)#
```

2. From the existing pool of syslog servers select the server or servers that will receive syslog messages.
3. Ensure that all selected servers are configured with the same value for the facility attribute.

   Allowable values are integers within the range 0 through 23.
4. Use the following command sequence to move to system-config Configuration Mode.

```
ACMEPACKET(syslog-config)# done
ACMEPACKET(syslog-config)# exit
ACMEPACKET(system-config)#
```

5. Use the ids-syslog-facility attribute to enable message transfer to specific syslog servers.

   The default value, -1, disables selective message transfer. To enable transfer to a designated syslog server or servers, enter the facility value (an integer within the range 0 through 23) that you confirmed or set in Step 3.

   The following example enables the transfer of IDS syslog messages to all servers with a facility value of 16.

```
ACMEPACKET(system-config)# ids-syslog-facility 16
ACMEPACKET(system-config)#
```

6. Use done, exit, and verify-config to complete this configuration.

TCA Reporting of Denied Entries

Users can construct a Threshold Crossing Alarm (TCA) which issues minor, major, and critical system alarms when the count of denied entries exceeds pre-configured values. For each issued alarm, the TCA also transmits an SNMP trap that reports the alarm state to remote SNMP agents.

1. Use the following command sequence to move to media-manager-config Configuration Mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# alarm-threshold
ACMEPACKET(alarm-threshold)#
```

2. Use the type attribute to specify the TCS type (deny-allocation for denied entries TCAs), the severity attribute to specify the criticality of the alarm, and the value attribute to specify the alarm threshold.

The following ACLI sequence defines three alarm thresholds (minor, major, and critical), although it is not necessary to configure all thresholds. Given the static deny allocation value of 32000 on a Net-Net 4500 or 16000 on a Net-Net 3800, you can determine what the percentage value maps to.

```
ACMEPACKET(alarm-threshold)# type deny-allocation
ACMEPACKET(alarm-threshold)# severity minor
ACMEPACKET(alarm-threshold)# value 80
ACMEPACKET(alarm-threshold)# done
ACMEPACKET(alarm-threshold)# type deny-allocation
ACMEPACKET(alarm-threshold)# severity major
ACMEPACKET(alarm-threshold)# value 90
ACMEPACKET(alarm-threshold)# done
ACMEPACKET(alarm-threshold)# type deny-allocation
ACMEPACKET(alarm-threshold)# severity critical
ACMEPACKET(alarm-threshold)# value 95
ACMEPACKET(alarm-threshold)# done
```

3. Use exit and verify-config to complete this configuration.

   After issuing a system alarm and accompanying SNMP trap, the TCA continues to monitor the number of denied entries. If the number of denied entries rises to the next threshold value, a new, and more severe, system alarm/SNMP trap is generated. If the number of denied entries falls below the current threshold level, and remains there for a period of at least 10 seconds, a new, and less severe system alarm/SNMP trap is generated.

Syslog Reporting of Denied Entries

Syslog reporting of endpoint demotions was introduced as part of IDS Phase 1 in S-C6.2.0. With IDS Phase 2, such syslog messages contain the last SIP message from the endpoint that caused the transition to the denied state. If the included SIP message increases the length of the syslog beyond 1024 bytes, the SIP message is truncated so that the syslog is no larger than 1024 bytes.

CPU Load Limiting

The transmission of IDS-related system alarms and SNMP traps is disabled when the CPU utilization rate surpasses a configured threshold percentage. When the threshold is exceeded, a syslog message (MINOR level) announces the termination of IDS reporting. No additional syslog messages or SNMP traps are generated until the CPU utilization rate falls below the configured threshold. The resumption of IDS reporting is announced by another syslog message, also issued at the MINOR level.

By default the CPU utilization rate is 90%. This value can be changed by the following ACLI command sequence

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)# options +load-limit=”80”
ACMEPACKET(sip-config)# done
```

Denied Endpoints

IDS Phase 2 provides a denied endpoint counter that includes SIP and MGCP (Media Gateway Control Protocol) endpoints. The global counter value is available via SNMP or HDR.

The global counter value is available to SNMP under APSYSMGMT-MIB -> acmepacketMgmt -> apSystemManagementModule -> apSysMgmtMIBObjects -> apSysMgmtMIBGeneralObjects.

apSysMgmtGeneralObjects Table (1.3.6.1.4.1.9148.3.2.1.1)		
Object Name	**Object OID**	**Description**
apSysCurrentEndptsDenied	1.3.6.1.4.1.9148.3.2.1.1.26	Global counter for current endpoints denied

The system HDR collection group contains a new reporting field, Current Deny Entries Allocated, which contains the value of the global endpoints denied counter.

Maintenance and Troubleshooting

show sipd acls

The show sipd acls command includes counters that track the number of endpoints demoted from trusted to untrusted and the number of endpoints demoted from untrusted to denied. For example:

```
ACMEPACKET# show sipd acls
...
ACL Operations          ---- Lifetime ----
                Recent      Total  PerMax
...
Trust->Untrust       0          1       1
Untrust->Deny        0          1       1
```

20

Transcoding

Introduction

Transcoding is the ability to convert between media streams that are based upon disparate codecs. The Oracle Enterprise Session Border Controller supports IP-to-IP transcoding for SIP sessions, and can connect two voice streams that use different coding algorithms with one another.

This ability allows providers to:

- Handle the complexity of network connections and the range of media codecs with great flexibility
- Optimize bandwidth availability by enforcing the use of different compression codecs
- Normalize traffic in the core network to a single codec
- Enact interconnection agreements between peer VoIP networks to use approved codecs

By providing transcoding capabilities at the network edge rather than employing core network resources for the same functions, the Oracle Enterprise Session Border Controller provides cost savings. It also provides a greater degree of flexibility and control over the codec(s) used in providers' networks and the network with which they interconnect.

In addition, placing the transcoding function in the Oracle Enterprise Session Border Controller and at the network edge means that transcoding can be performed on the ingress and egress of the network. The Oracle Enterprise Session Border Controller transcodes media flows between networks that use incompatible codecs, and avoids back-hauling traffic to a centralized location, alleviating the need for multimedia resource function processors (MRFPs) and media gateways (MGWs) to support large numbers of codecs. This maximizes channel density usage for the MRFPs and MGWs so that they can reserve them for their own specialized functions.

Transcoding Hardware

A Transcoding NIU (Network Interface Unit) provides the DSP resources that enable the Oracle Enterprise Session Border Controller's transcoding feature. Different platforms' transcoding NIUs can accept different numbers of transcoding modules.

- The AP3820 and AP4500 can accept 1 to 12 transcoding modules to provide increasing transcoding capacity.

Transcoding Capacity

Transcoding capacity depends on the following:

- Codecs used for transcoding
- Number of transcoding modules installed in the system. Capacity scales linearly with each extra transcoding module installed.

Transcodable Codec Details

The following table lists the supported codecs, bit rates, RTP payload number, default ptime, and supported ptimes.

Codec	Supported Bit Rate (kbps)	RTP Payload Type	Default Ptime (ms)	Supported Ptime (ms)
G.711 PCMU	64	0	20	10, 20, 30, 40, 50, 60
G.711 PCMA	64	8	20	10, 20, 30, 40, 50, 60
G.722	48, 56, 64	9	20	10, 20, 30, 40
G.723.1	5.3, 6.3	4	30	30, 60, 90
G.726	16, 24, 32, 40	2, 96-127	20	10, 20, 30, 40, 50
iLBC	13.33	96-127	20	20, 30, 40, 60
	15.2	96-127	30	20, 30, 40, 60
G.729/A/B	8	18	20	10, 20, 30, 40, 50, 60, 70, 80, 90
AMR	4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2, 12.2	96-127	20	20, 40, 60, 80, 100
AMR-WB (G. 722.2)	6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, 23.85	96-127	20	20, 40, 60, 80, 100
GSM FR	13	3	20	20
T.38	4.8, 9.6, 14.4	N/A		10, 20, 30

T.38 FAX Support

This release supports T.38 FAX relay (Version 0) conversion to T.30 over G.711 and supports FAX modulation schemes up to 14400 kbps V.17. The initial release does not support V.34 modulation.

Session Licensing

AMR/AMR-WB and EVRC/EVRC-B audio codecs require extra licenses while all other codecs are implicitly licensed by installation of a transcoding NIU. These codecs are under royalty agreements which necessitates their own special licenses.

Licenses for AMR/AMR-WB and EVRC/EVRC-B transcoding sessions are installed by groups of 25.

Software-based transcoding

The Oracle Enterprise Session Border Controller supports media transcoding on COTS and VM based systems. The transcoding capacity is limited to 100 sessions.

Transcoding is the process of converting voice audio streams from one encoding format (codec) to another. In addition to conversion between codecs, the Oracle Enterprise Session Border Controller can also reframe compressed audio streams from one packet size to another (e.g. 10ms G.729 reframed to 30ms G.729) according to packetization times specified in session establishment.

The Oracle Enterprise Session Border Controller may then convert between any supported codecs and frame size combination to another supported codec and frame size combination. The following are the supported codecs/bit rate, and frame sizes (packetization interval):

Codec	Bit rate	Packetization intervals
G.711 (PCMU/PCMA)	64 kbps	10, 20, 30, 40, 50, 60 ms
G.729/A/B	8 kbps	10, 20, 30, 40, 50, 60 ms

Software-based transcoding is configured identically to hardware-based transcoding, and is invoked when codec policies are configured but no transcoding hardware is recognized in the system. The **dtmf-in-audio** parameter is absent from the **codec-policy** configuration element because DTMF detection/generation is unsupported for COTS and VM based systems.

The following items are also not supported in systems with software-based transcoding only:

- T.38 Fax IWF
- In-band DTMF digit detection/generation
- QoS marking
- IPv4 to IPv6 interworking for transcoded calls

Codec Royalty Licensing

Use of the G.729 Requires a license due to royalty agreements. Licenses are installed on the system in groups of 25 sessions.

Software-based transcoding alarms and traps

SNMP Traps

The apSysMgmtGroupTrap trap is sent with the MIB OID apSysXCodeG729Capacity to alert you of high G.729 Royalty codec usage. This MIB object is defined in ap-smgmt.mib. It is sent when utilization rises above 95% of licensed capacity. It is cleared when utilization falls below 80% of licensed capacity. The MIB object appears as:

```
apSysXCodeG729Capacity OBJECT-TYPE
        SYNTAX          SysMgmtPercentage
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
                "The percentage of licensed G729 transcoding utilization"
        ::= { apSysMgmtMIBGeneralObjects 35 }
```

Alarms

The G729 transcoding utilization alarm is triggered when utilization rises above 95% of licensed capacity. It is cleared when utilization falls below 80% of licensed capacity. The alarm appears as follows on the ACLI:

```
ID      Task      Severity      First Occurred          Last Occurred
131159  527739792       6       2011-10-11 10:11:49     2011-10-11 10:11:49
Count   Description
1       G729 Transcoding capacity at  97 (over threshold of 95)
```

Debugging log files

The log.media log file records host based transcoding events based upon logging level.

Transcoding Configuration

The Oracle Enterprise Session Border Controller performs transcoding functions— allowing the entities with incompatible codecs to communicate with each other— between two call legs. The two endpoints can be located in one or two realms or networks. The Oracle Enterprise Session Border Controller decides to transcode a call by evaluating messages in the SDP offer-answer transaction with respect to system configuration. An SDP offer can be in a SIP message such as an INVITE or a reINVITE, and contains information about the codecs the offerer would like

to use. The answerer answers the SDP offer with its own set of supported codecs. reINVITEs are treated as new negotiations, with respect to the actual SDP offerer and answerer. The Oracle Enterprise Session Border Controller can manipulate an SDP message by reordering codec preference, and by adding and deleting codecs.

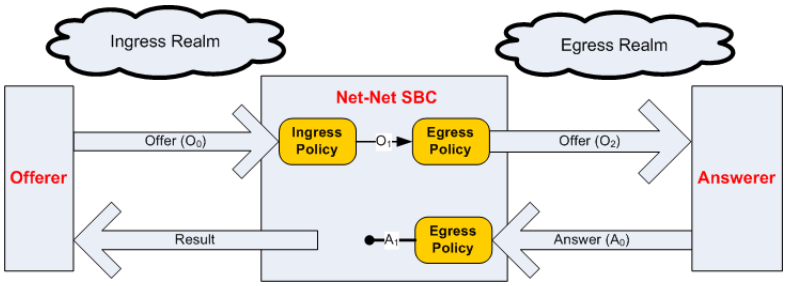
Transcoding Processing Overview

Transcoding processing is viewed in terms of the ingress and egress realms. The ingress realm is where the SDP offer is received by the Oracle Enterprise Session Border Controller. The egress realm is where the SDP offer is sent, and where the SDP answer is expected to be received from (i.e., the answerer's realm). A call is defined as transcodable if an egress or ingress policy exists for the session and if the session is not subject to media release, as specified in the realm configuration.

To understand the details of transcoding processing, refer to the following diagram. An SDP offer, O0, is received by the Oracle Enterprise Session Border Controller in the ingress realm. The ingress codec policy is applied here and the SDP offer is transformed into O1. O1 is then passed to and processed by the egress codec policy. This SDP message is then forwarded to the answerer as O2. The answerer replies with A0 to the Oracle Enterprise Session Border Controller, which is subjected to the egress codec policy again and transformed into A1.

When policy dictates not to transcode the call, the Result SDP sent back to the offerer is based on the common codecs shared between A1 and O1. The Oracle Enterprise Session Border Controller first constructs the list of codecs that are present in both in O1 and A1. Then, the Oracle Enterprise Session Border Controller maintains the codec order from A1 for the Result as it is sent to the offerer.

When policy dictates to transcode the call, the top transcodable codec in O1 is used in the ingress realm and the top non-signaling codec in A1 is used in the egress realm.



Defining Codec Policies

The following definitions are required for understanding transcoding processing:

DTMFable Codecs—Uncompressed codecs that are capable of properly transmitting a DTMF waveform. The only codecs designated as DTMFable are PCMU and PCMA.

FAXable Codecs—Uncompressed codecs that are capable of properly transmitting a T.30 waveform. The only codecs designated as FAXable are PCMU and PCMA.

Signaling Codecs—Non-audio codecs that are interleaved into a media stream but cannot be used on their own. The only codecs designated as Signaling Codecs by the Oracle Enterprise Session Border Controller are telephone-event and CN (comfort noise).

Disabling an m= line—This is in reference to an m= line in an SDP message. It means setting the m= line's port to 0 (RFC 3264). Enabling an m= line means it has a non-zero port. The m= line's mode attribute (sendrecv/inactive/rtcponly, etc) is not considered.

A codec policy is created by configuring the following information:

- Which Codecs are allowed and which are denied in a realm.
- Which Codecs should be added to the SDP m= lines for an egress realm.
- The preferred order of codecs to indicate in an SDP m= line.
- The packetization time which should be enforced within a realm.

Ingress Policy

Incoming SDP is first subject to the ingress codec policy. If no codec policy is specified in the realm config for the ingress realm, or the m= lines in the SDP offer are disabled (by a 0 port number), the SDP is transformed to O1 unchanged.

The ingress codec policy first removes all un-allowed codecs, as configured in the allow-codecs parameter by setting their port to 0 or removing the codecs from a shared m-line. For example, if two codecs share an m-line and one of them is un-allowed, the resulting m-line will not include the un-allowed codec and its attribute lines will be removed. If a single codec is used, the resulting m-line will include the codec, but its port will be set to 0 and its attribute lines will remain. Next, the remaining codecs are ordered with the order-codecs parameter. Ordering is when the codec policy rearranges the codecs in the SDP m= line. This is useful to suggest the codec preferences to impose within the egress realm. O1 is then processed by the egress codec policy after a realm is chosen as the destination.

In practical terms, the ingress policy can be used for filtering high-bandwidth codecs from the access realm. It can also be used for creating a suggested, prioritized list of codecs to use in the ingress realm.

Egress Policy

The Oracle Enterprise Session Border Controller applies egress codec policy to the SDP that has already been processed by ingress policy. The egress policy is applied before the SDP exits the system into the egress realm. If no egress codec policy is defined, or the SDP's m= lines are disabled (with a 0 port), the SDP is passed untouched from the ingress policy into the egress network.

The egress codec policy first removes all un-allowed codecs in the allow-codecs parameter (<codec>:no). Codecs on the add-codecs-on-egress list are not removed from the egress policy regardless of the how the allow-codecs parameter is configured. If the result does not contain any non-signaling codecs, the ptime attribute is removed from the SDP. Codecs not present in O1 that are configured in the add-codecs-on-egress parameter are added to the SDP, only if O1 contains one or more transcodable codecs.

Note: Transcoding can only occur for a call if you have configured the add-codecs-on-egress parameter in an egress codec policy.

If codecs with dynamic payload types (those between 96 and 127, inclusive) are added to the SDP, the lowest unused payload number in this range is used for the added codec.

The following rules are also applied for egress policy processing:

If O1 contains at least one transcodable codec, the codecs listed in the Egress policy are added to the SDP.

- telephone-event, as configured in add-codecs-on-egress will only be added if O1contains at least one DTMFable codec.
- T.38, as configured in add-codecs-on-egress will only be added if there is no T.38 and there is at least one FAXable codec (G711Fall Back (FB)) in O1. T.38 will then be added as a new m= image line to the end of SDP.

  If G711FB is not allowed in the egress policy, the Oracle Enterprise Session Border Controller disables the m= line with the FAXable codec. Otherwise if G711FB is allowed, pass it through the regular offer processing allowing/adding only FAXable codecs.
- G711FB, as configured in add-codecs-on-egress will only be added if there is no G711FB and there is T.38 in O1. G711FB will then be added as a new m= audio line to the end of SDP.

  If T.38 is not allowed in the egress policy, the Oracle Enterprise Session Border Controller disables the m= image line. Otherwise if T.38 is allowed, pass it through the regular offer processing.

If the result of the egress policy does not contain any non-signaling codecs, audio or video, the m= line is disabled, by setting the port number to 0.

The m= line is next ordered according to rules for the order-codecs parameter.

Finally, all attributes, a= lines, ptime attribute, and all other unrecognized attributes are maintained from O1. Likewise, appropriate attributes for codecs added by the add on egress parameter are added to SDP. Finally, rtpmap and fmtp parameters are retained for codecs not removed from the original offer. The result of all this is O2, as shown in the overview diagram.

In practical terms, codec policies can be used to normalize codecs and ptime in the core realm where the network conditions are clearly defined.

Codec policies can also be used to force the most bandwidth-conserving codecs anywhere in the network.

Post Processing

If any errors are encountered during the Ingress and Egress policy application, or other violations of RFC3264 occur, the call is rejected. If O2 does not contain any enabled m= lines at the conclusion of the initial call setup, the call is rejected. If O2 does not contain any enabled m= lines at the conclusion of a reINVITE, the reINVITE is rejected and the call reverts back to its previous state.

Codec Policy Definition

Codec policies describe how to manipulate SDP messages as they cross the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller bases its decision to transcode a call on codec policy configuration and the SDP. Each codec policy specifies a set of rules to be used for determining what codecs are retained, removed, and how they are ordered within SDP.

Syntax

The following parameters are used to create a codec policy. Their syntax is described inline.

allow-codecs

allow-codecs—The allow-codecs parameter configures the codecs that are allowed and/or removed from the SDP. A blank list allows nothing, * allows all codecs, none removes all codecs, the :no designation blocks the specific codec or class of media, and the :force designation is used to remove all non-forced codecs.

The allow-codecs parameter is configured in the following way:

- <codec>:no—blocks the specific codec
- *—allow all codecs.
- <codec>:force—If any forced codec is present in an SDP offer, all non-forced codecs are stripped from the m-line.
- audio:no—audio m= line is disabled
- video:no—video m= line is disabled

For example, if you configure PCMU in the allow-codecs parameter, the PCMU codec, received in an SDP message is allowed on to the next step of transcoding processing, and all other codecs are removed.

The order of precedence is for removing codecs according to codec policy is:

1. <codec>:no—Overrides all other allow-codecs parameter actions.
2. audio:no / video:no. An allow-codecs line like “allow-codecs PCMU audio:no” disables the PCMU m= line because audio:no has a higher precedence than the specific codec.
3. <codec>:force
4. <codec> Specific codec name and those codecs configured in the add-codecs-on-egress list.
5. * has the lowest precedence of all flags. For example "allow-codecs * PCMU:no" allows all codecs except PCMU.

order-codecs

order-codecs—The order-codecs parameter is used to re-order the codecs in the m= line as the SDP is passed on to the next step. This parameter overwrites the order modified by the add-codecs-on-egress command, when relevant. The following is valid syntax for this parameter:

- <blank>—Do not re-order codecs
- *—You can add a <codec> before or after the * which means to place all unnamed codecs before or after (the position of the *) the named codec. For example:

- <codec> *—Puts the named codec at the front of the codec list.
- * <codec>—Puts the named codec at the end of the codec list.
- <codec1 > * <codec2>—Puts <codec1> first, <codec2> last, and all other unspecified codecs in between them.
- <codec>—When the * is not specified, it is assumed to be at the end.

Any codec name is allowed in the order-codecs parameter, even those not defined or not transcodable. An * tells the order-codecs parameter where to place unspecified codecs with respect to the named codecs. Refer to the examples below.

- <blank>—do not reorder m= line
- PCMU *—Place PCMU codec first, all others follow
- * PCMU—Place PCMU codec last, all others proceed PCMU
- G729 * PCMU—Place G729 codec first, PCMU codec last, all others remain in between
- PCMU—If * is not specified, it is assumed to be at the (PCMU *).

Add on Egress

add-codecs-on-egress—This parameter adds a codec to the SDP's m= line only when the codec policy is referenced from an egress realm (except in one 2833 scenario). Codecs entered for this parameter are added to the front of the m= line. Signaling codecs are added to the end of the m= line.

Transcoding can only occur if this parameter is configured. There is a special case for 2833 support where the add-codecs-on-egress parameter is configured for an ingress realm. See *RFC 2833 Scenario 2* for details.

Packetization Time

packetization-time—This parameter specifies a media packetization time in ms to use within the realm referencing this codec policy. Packetization time You must also enable the force ptime parameter to enable transrating in conjunction with configuring the packetization time. See the *Transrating* section for more information.

Answer Processing and Examples

Unoffered Codec Reordering

According to RFC 3264, the answerer can add codecs that were not offered to the Answer. The answerer may add new codecs as a means of advertising capabilities. RFC 3264 stipulates that these unoffered codecs must not be used. The RFC does not dictate where in the m= line these codecs can appear and it is valid that they may appear as the most preferred codecs.

In order to simplify the answer processing, the Oracle Enterprise Session Border Controller moves all unoffered codecs in A0 to the back of the SDP answer before any other answer processing is applied.

Non-transcoded Call

The decision to transcode is based on the top non-signaling codec in A1. If the top A1 codec is present in O1, the call proceeds, non-transcoded. This is the rule for non-signaling codecs (i.e., not RFC 2833 nor FAX).

Transcoded Call

The following two conditions must then be met to transcode the call's non-signaling media:

- The top A1 codec is not present in the O1 m= line
- The top A1 codec was added by the egress policy

If these rule are met, the Oracle Enterprise Session Border Controller will transcode between the top A1 codec and the top transcodable, non-signaling O1 codec.

Voice Transcoding

The following examples use the ingress and egress codec policies listed at the top of each scenario. The examples use changing SDP offers and answers, which contribute to unique results, per example. The effects of the SDP offers and answers are explained in each example.

Voice Scenario 1

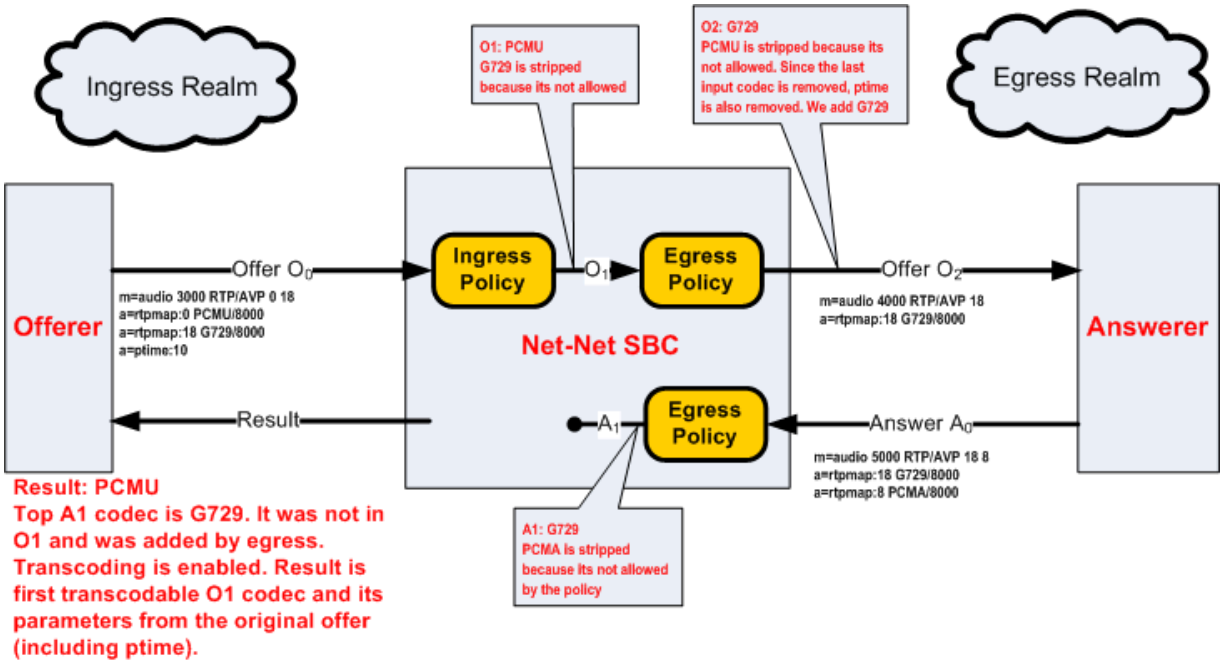
The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow-codecs	PCMU GSM	allow-codecs	G729 GSM G722
add-codecs-on-egress	PCMU	add-codecs-on-egress	G729
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

Note: The codec in the ingress policy's add-codecs-on-egress parameter has no effect in the following examples. Its presence would have an effect if a reINVITE was initiated from egress realm, effectively reversing the roles of the codec policies.

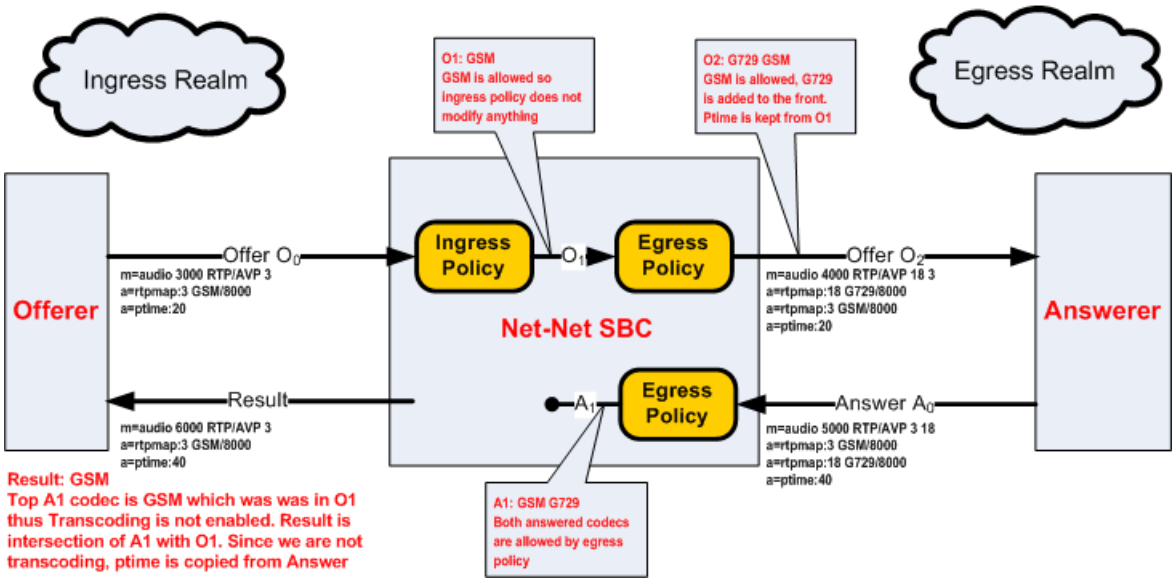
1. In the following diagram, PCMU and G729 are offered. Ingress policy removes G729 and allows PCMU. The egress policy adds G729 and strips PCMU from offered SDP and forwards it on to the answerer (ptime is also removed because the last codec was removed).

   The SDP answer agreed to use G729 and adds PCMA. The egress policy then strips PCMA from the SDP answer. At this point, the top codec in A1, G729 is checked against O1. Since G729 is not present in O1, it is transcoded to PCMU.

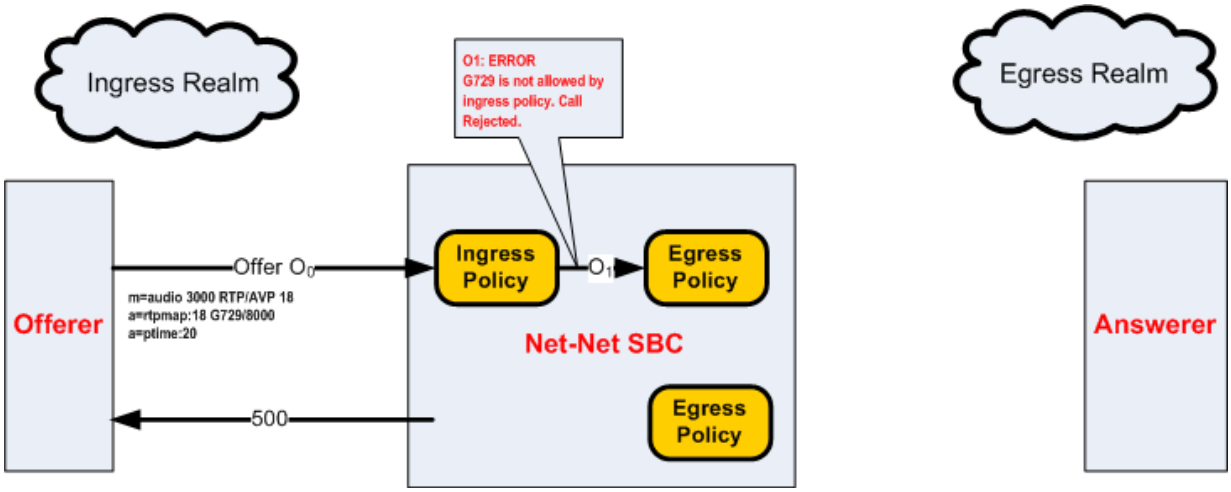


2. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from GSM and sends this to the answerer as O2.

The SDP answer agrees to use G729 and GSM, but prioritizes GSM. The egress policy allows both codecs through, unchanged. Because A1 and O1 both have GSM, it is used for the non-transcoded call. Ptime is copied from A0 to the result.

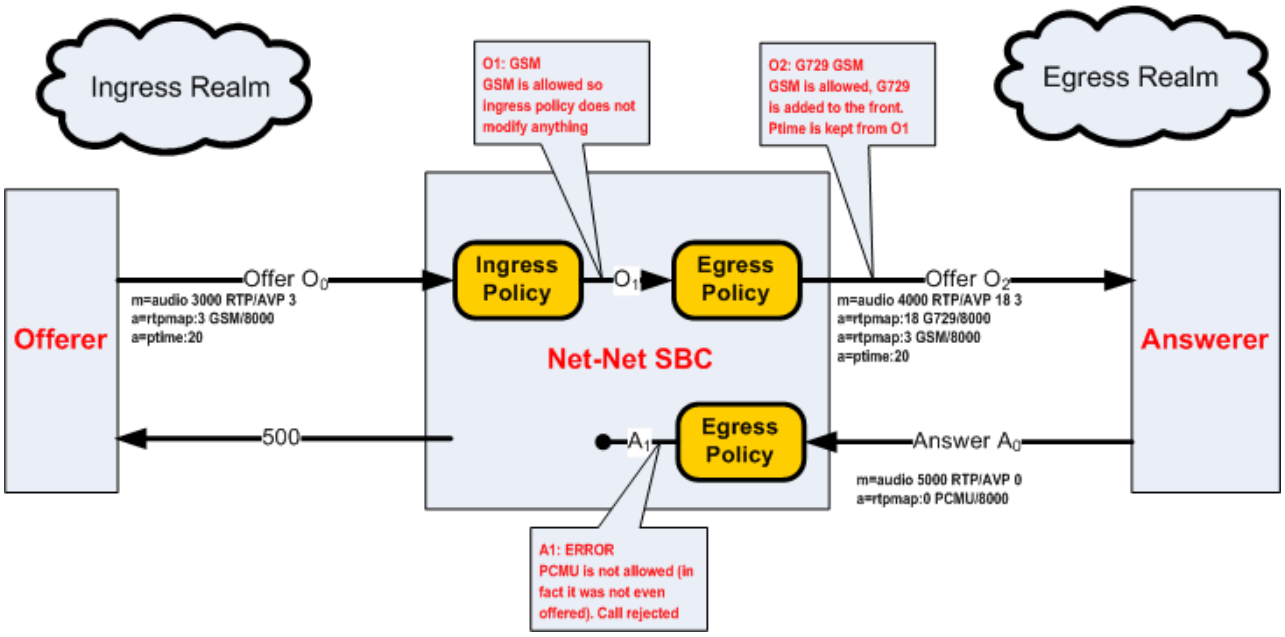


3. In the following diagram, G729 in the original SDP offer. Because once G729 is removed, no non-signaling are left in O1, thus the call is rejected.



4. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from O1 and sends this to the answerer as O2.

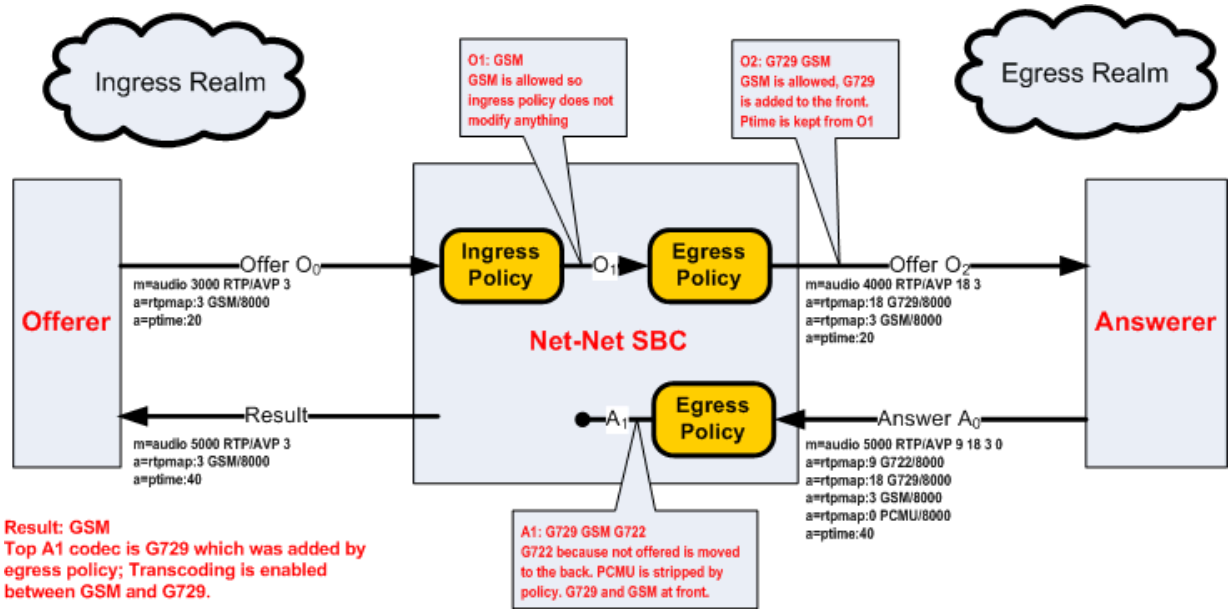
   The SDP answer states that the answerer wants to use PCMU. This is a violation of the RFC3264. Therefore the call is rejected.



In this example, when the negotiation fails, the Oracle Enterprise Session Border Controller sends a 500 message to the offerer and a BYE message to the answerer.

5. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from O1 and sends this to the answerer as O2.

   The SDP answer replies with G722 G729 GSM and PCMU. PCMU is stripped by policy, G722 is moved to the back of the answer because it was not offered. The top A1 codec was not in O1, and was added by egress policy, therefore the call is transcoded between GSM and G729.



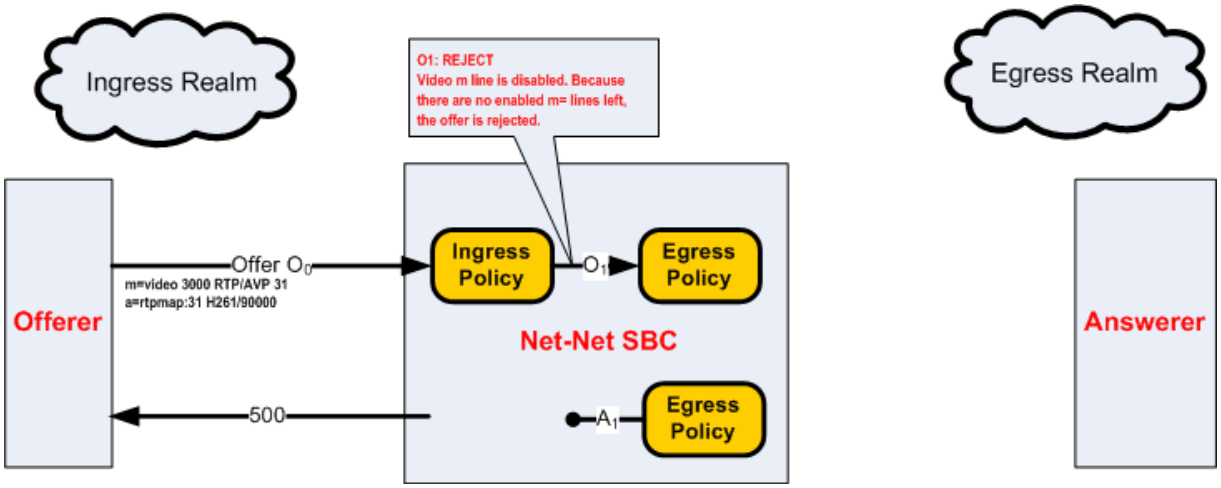
Voice Scenario 2

The following ingress and egress policies are used for scenario 2.

Ingress Policy		Egress Policy	
allow-codecs	Video:no PCMU:force * PCMA:force	allow-codecs	* PCMA:no

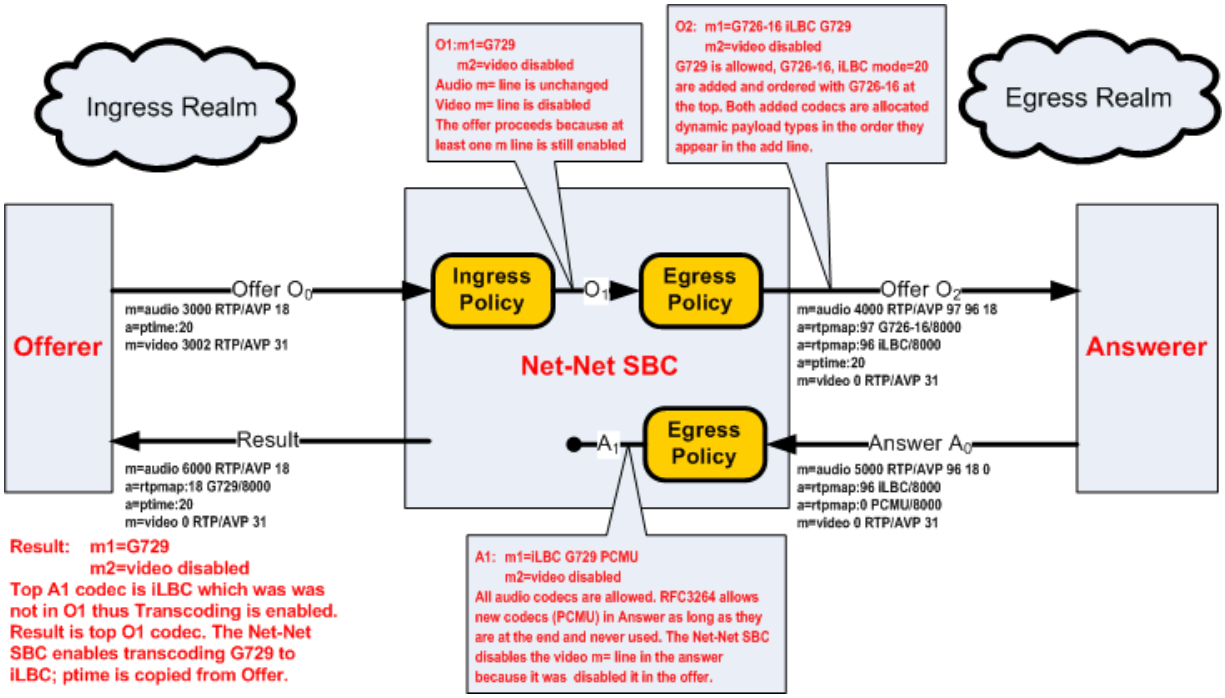
Ingress Policy		Egress Policy	
add-codecs-on-egress		add-codecs-on-egress	iLBC G726-16
order-codecs		order-codecs	G726-16 * PCMU
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, a video m= line is offered. The ingress policy disables the video m= lines. With no enabled m= lines left, the call is rejected.



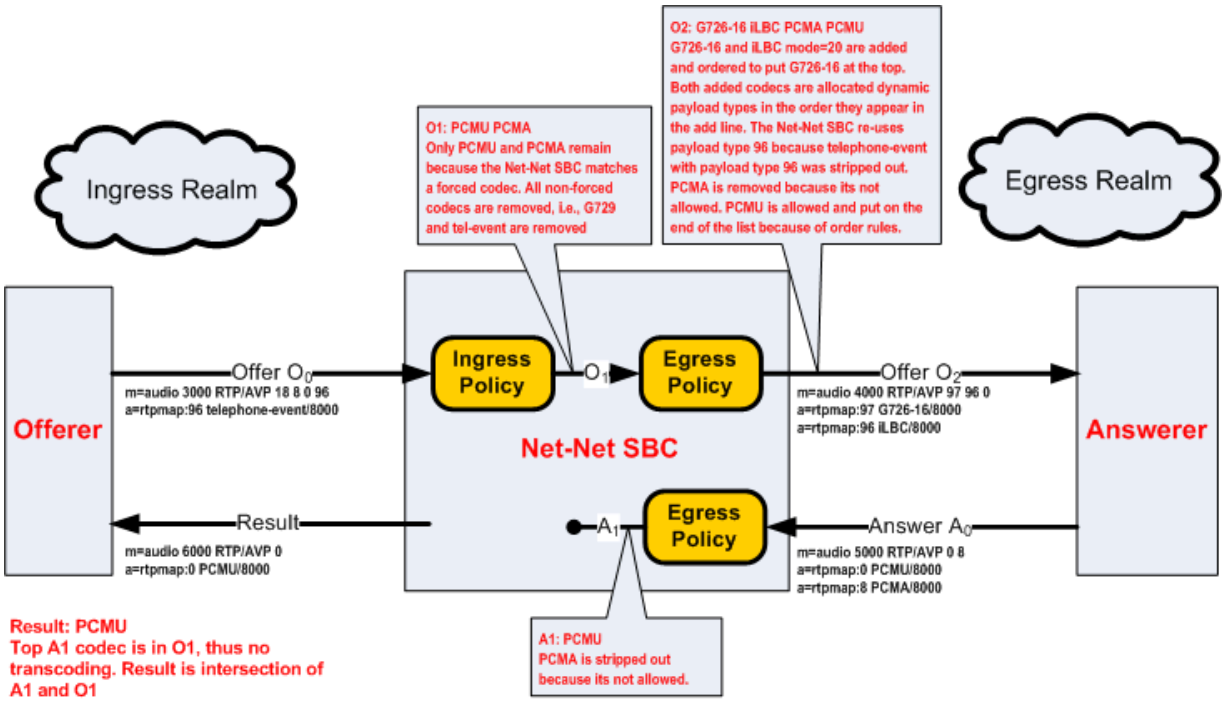
2. In the following diagram, G729 and video are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order-codecs parameter. The ptime is maintained between O0 and O2. Both added codecs are allocated dynamic payload types in the order they appear in their m= line. A disabled Video m= line is passed on to the answerer.

   The SDP answer agreed to use iLBC, G729, and adds PCMU, and reorders them as stated. The disabled video m= line is maintained. At this point, the top codec in A1, iLBC is used and transcoded with the top codec in O1, G729.



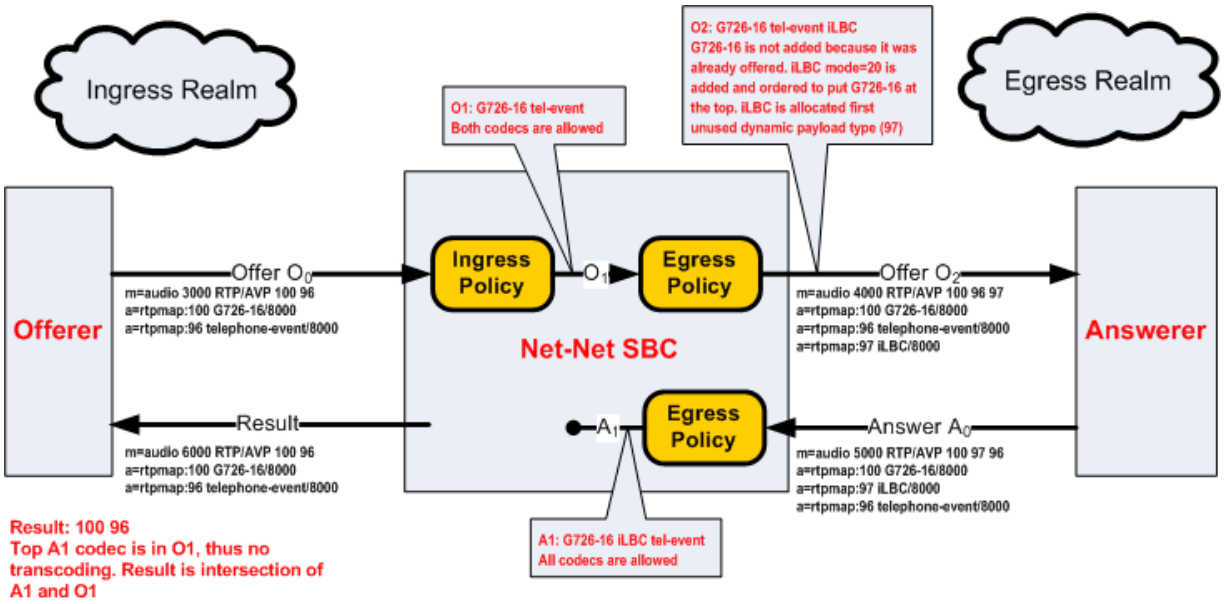
3. In the following diagram, G729 and video are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order-codecs parameter. The ptime is maintained between O0 and O2. Both added codecs are allocated dynamic payload types in the order they appear in their m= line.

   The SDP answer only wants to use PCMU and PCMA. The egress policy removes PCMA and passes only PCMU to the offerer. Because PCMU was in O1 and is now the only codec in A1, it is used, and no transcoding is used between the endpoints.



4. In the following diagram, G726-16 and telephone-event are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows both codecs. The egress policy adds iLBC, and then orders the codecs according to the order-codecs parameter.

The SDP answer agreed to use all codecs, but reorders them with G726-16 in the top position. Because G726-16 is the top codec in A1, and it is also present in O1, it is used for this call without any transcoding.

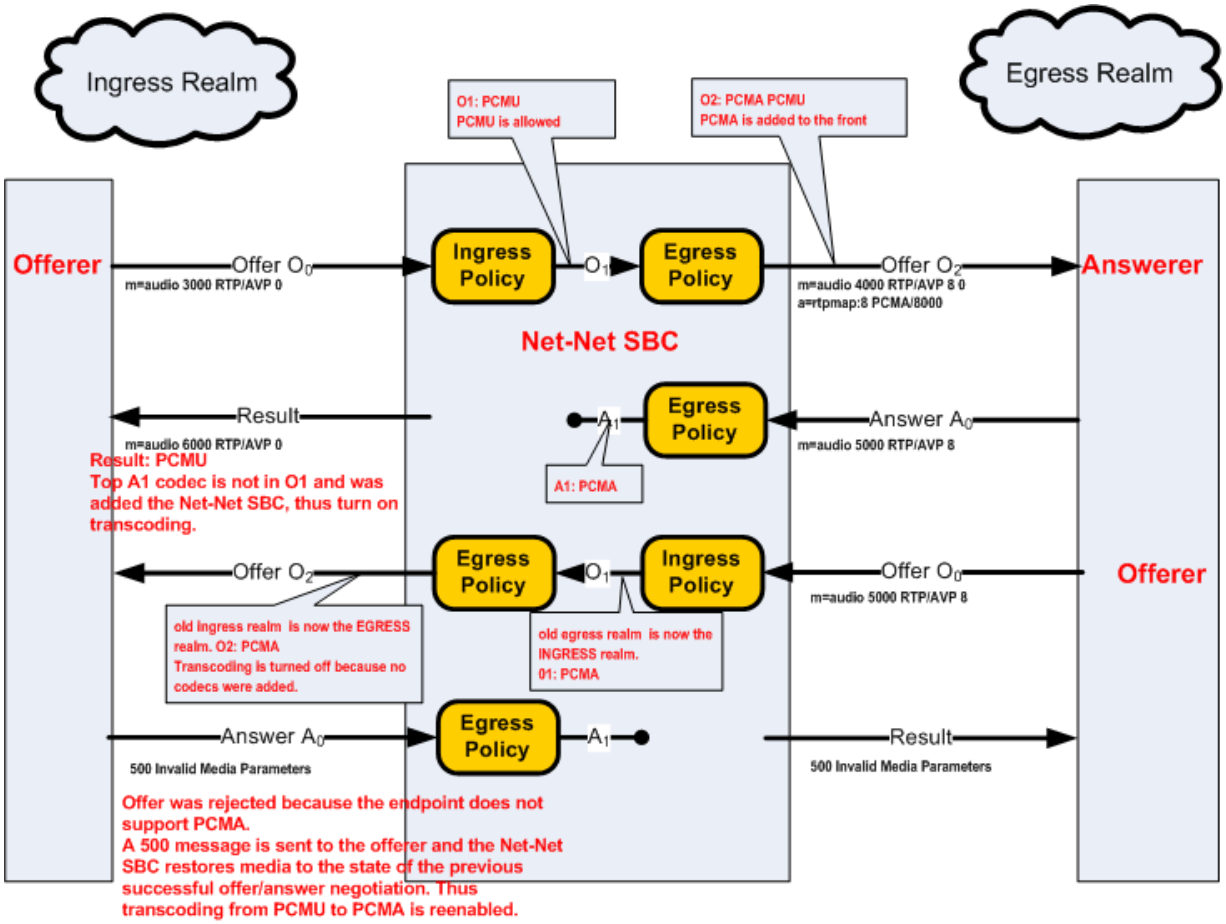


Voice Scenario 3

Voice scenario 3 involves reINVITEs. The following ingress and egress policies are used for scenario 3.

Ingress Policy		Egress Policy	
allow-codecs	PCMU G729	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	PCMA
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

In the following diagram, the answerer sends a reINVITE after a previous transcoding session was established. The original offerer and answerer swap roles. The new offerer rejects the SDP offer and the call reverts to the state negotiated in the original SDP negotiation.



RFC 2833 Transcoding

RFC 2833 defines an RTP payload that functions interchangeably with DTMF Digits, Telephony Tones and Telephony Signals. The Oracle Enterprise Session Border Controller can monitor audio stream for in-band DTMF tones and then can convert them to data-based telephone-events, as sent in RFC2833 packets. This section explains how the Oracle Enterprise Session Border Controller transcodes between these RTP-based telephone events and in-band DTMF tones carried by G711. DTMF tones can only be transported in non compressed codecs. The Oracle Enterprise Session Border Controller supports two DTMFable non-compressed codecs: PCMU (G711µ) and PCMA (G711A).

Note: The following line is added to SDP whenever telephone-event is added on egress: a=fmtp:101 0-15

The following two scenarios describe when telephone-event to DTMF transcoding takes place:

RFC 2833 Scenario 1

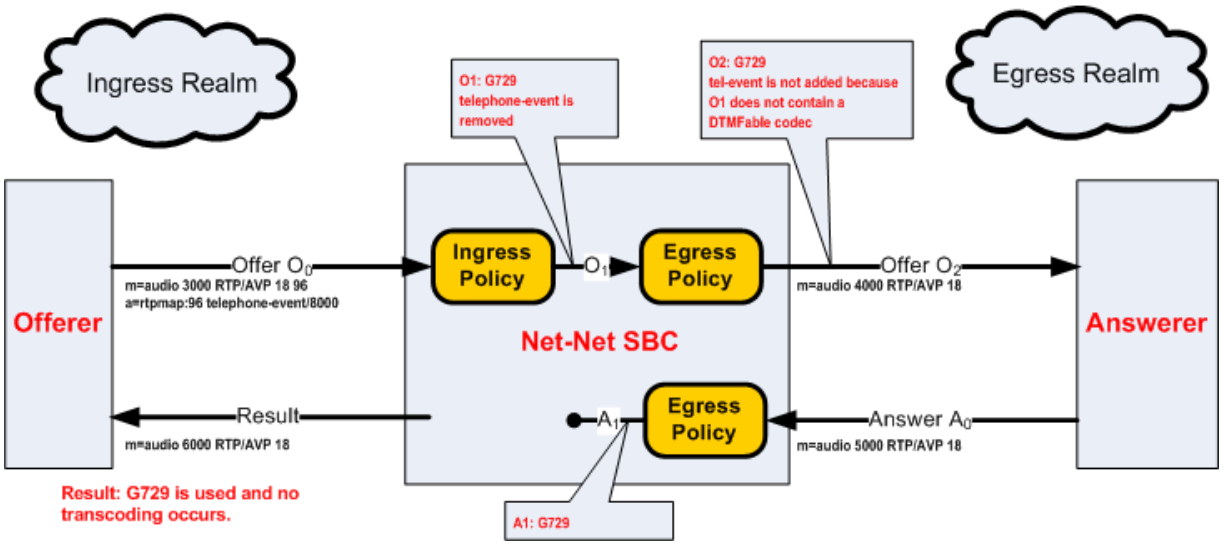
The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow-codecs	* telephone-event:no	allow-codecs	* PCMA:no
add-codecs-on-egress		add-codecs-on-egress	telephone-event
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled

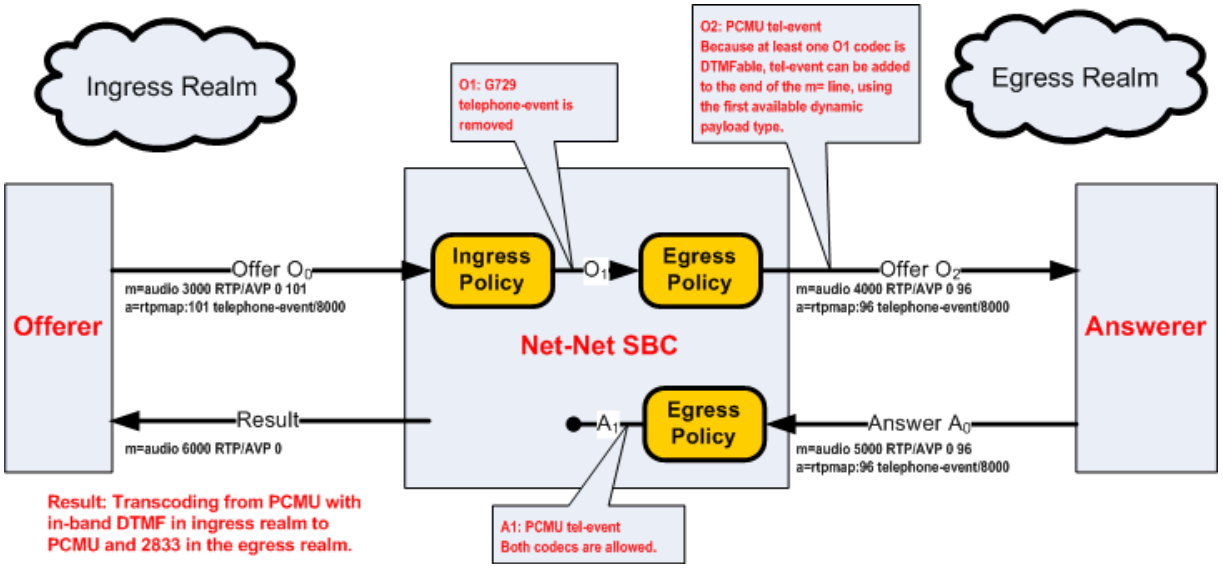
Ingress Policy		Egress Policy	
packetization-time		packetization-time	
dtmf-in-audio	preferred	dtmf-in-audio	preferred

1. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was not added by the egress policy because the remaining audio codec in O1 was not DTMFable. G729 was the only codec forwarded on to the answerer.

   The SDP answer agreed to use the remaining audio codec, G729. A0 is unaltered by egress policy, and forwarded as the Result to the offerer. Therefore, G729 is used in both the ingress and egress realms, the call does not support RFC 2833, and the call is not transcoded.

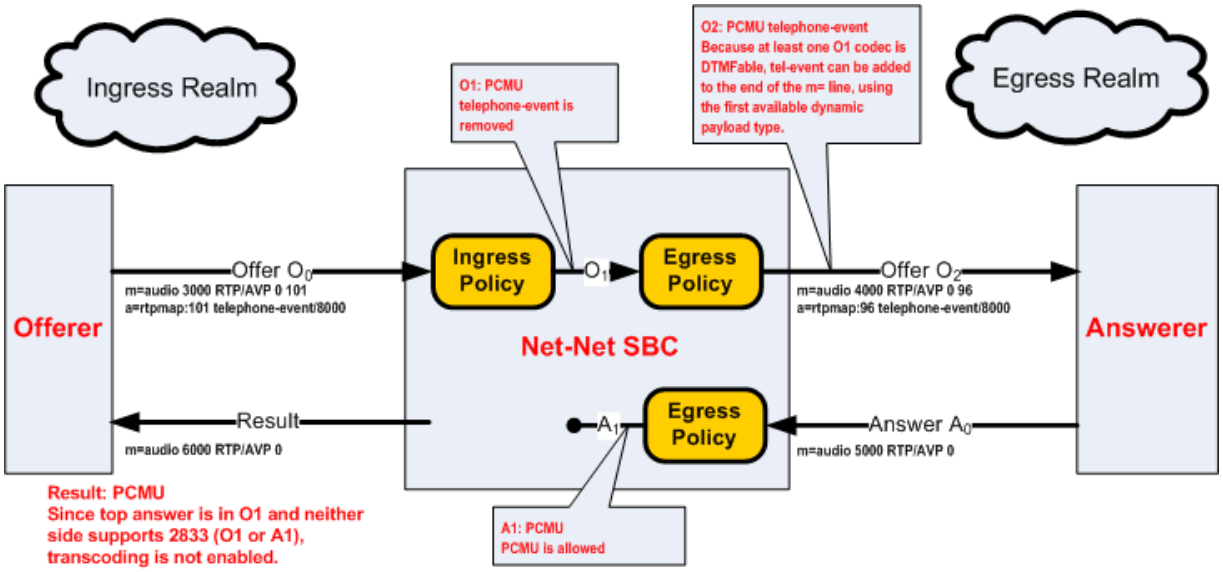


2. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was added by the egress policy because the remaining audio codec in O1 was DTMFable. PCMU and telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.



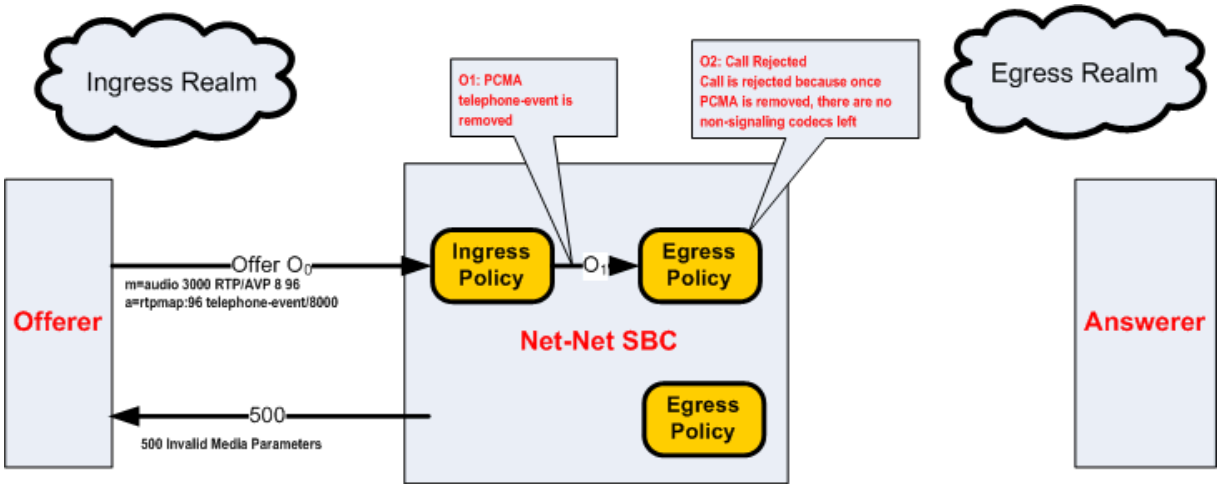
This case illustrates when the answerer supports audio and RFC 2833, but the offerer supports audio with inband DTMF. The Oracle Enterprise Session Border Controller transcodes between RFC2833 in the egress realm to in-band DTMF on the ingress realm.

3. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event is added by the egress policy because the remaining audio codec in O1 was DTMFable. PCMU and telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.



The SDP answer only agreed to use PCMU. When A0 reaches the egress policy, it is passed along through the Oracle Enterprise Session Border Controller to the offerer. Because telephone-event was not answered by the answerer and not supported in O1, it can't be used. Transcoding is therefore not used for this call.

4. In the following diagram, telephone event was offered by the offerer and was stripped by ingress policy. Since PCMA was also stripped by the egress policy, leaving no non-signaling codecs, the call is rejected. A 500 message is sent back to the offerer.



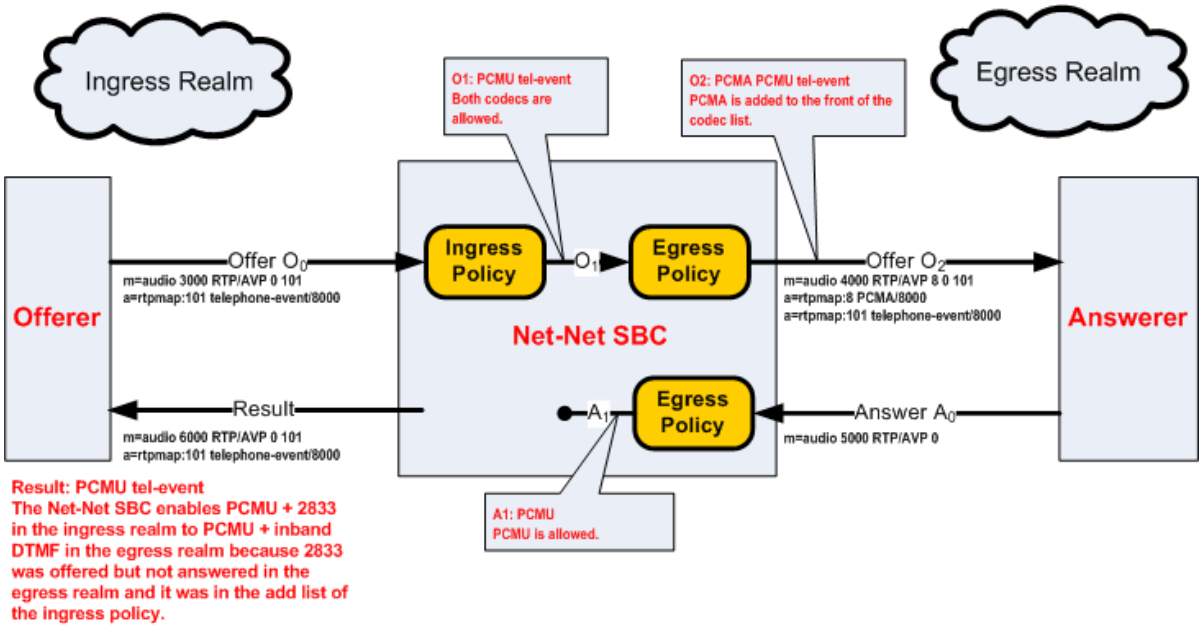
RFC 2833 Scenario 2

The following ingress and egress policies are used for RFC2833 scenario 2.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress	telephone-event	add-codecs-on-egress	PCMU
order-codecs		order-codecs	

Ingress Policy		Egress Policy	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	
dtmf-in-audio	preferred	dtmf-in-audio	preferred

1. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O1, and PCMA is added as it is sent to the answerer. The SDP answer, A0 disables all codecs but PCMU.

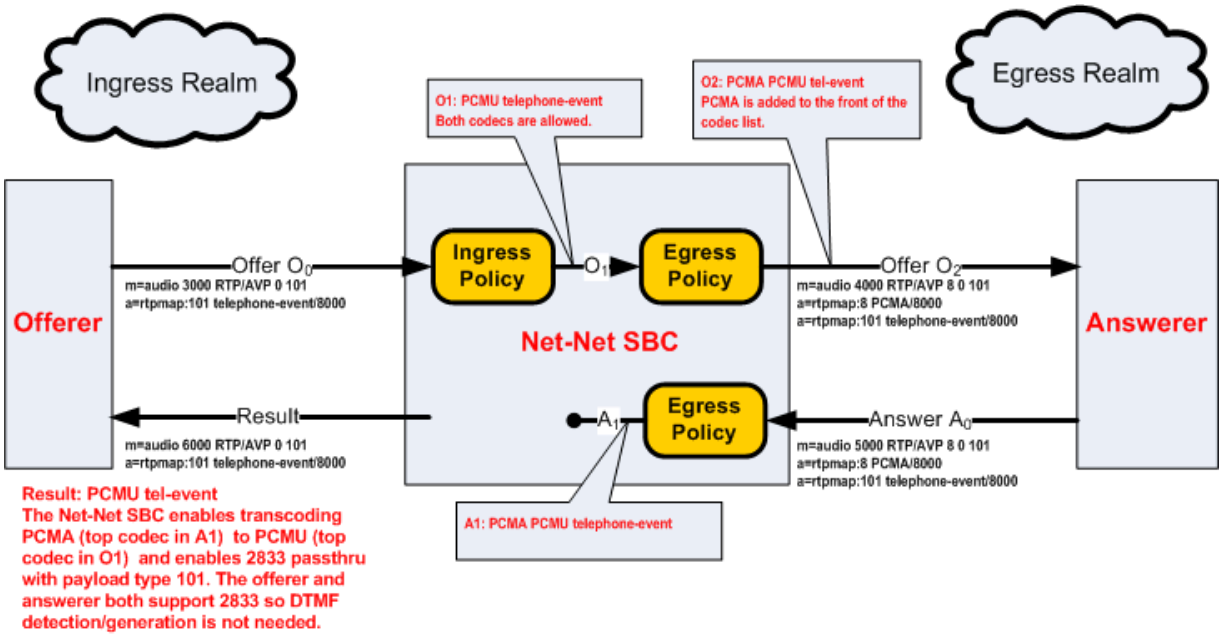


The Oracle Enterprise Session Border Controller adds telephone-event to the result because it is listed on the ingress policy's add-codecs-on-egress parameter and present in the offerer's SDP.

Note: This is the only time the add list of an ingress policy is utilized as a check.

The result SDP includes PCMU and telephone-event in the ingress realm, which is transcoded to PCMU with in-band DTMF in the egress realm.

2. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O1, and PCMA is added as it is sent to the answerer. The SDP answer supports all three codecs offered, with PCMA added on top.



The answerer responds with PCMA as the preferred codec in A0. The Oracle Enterprise Session Border Controller compares A1 to O1 to make the transcoding decision. PCMA is the top codec in A1 and is transcoded to PCMU, the top codec in O1. Also, because telephone-event is supported by both sides of the call, it is passed through without any transcoding necessary.

This case illustrates when both endpoints are capable of sending and receiving telephone-event. Regardless of whether the audio portion of the call is transcoded, the telephone-event messages are passed through the system untouched, thus not requiring transcoding resources. This is known as telephone-event pass-through.

FAX Transcoding

FAXes are transmitted in a call as either T.30 and T.38 media. T.30 FAX is binary in-band media carried over G.711. The Oracle Enterprise Session Border Controller can transcode between T.38 and a faxable codec. The supported faxable codecs are PCMU and PCMA.

T.30 can only be transported in non-compressed codecs. The two non-compressed codecs supported by the Oracle Enterprise Session Border Controller are PCMU (G711µ) and PCMA (G711A). If a transcoding realm does not support an uncompressed codec, T.30 can not be supported in that realm. Alternatively, G711FB may be allowed specifically for FAX only.

The Oracle Enterprise Session Border Controller uses an internal codec called G711FB (G711 - Fall Back) that is an umbrella codec of all FAXable codecs. G711FB will default to PCMU for the purpose of offering a faxable codec. You can remap G711FB to PCMA by configuring the media-profile for it appropriately. G711FB's only use is for FAX transcoding.

FAX transcoding is triggered when you configure the add on egress parameter with either T.38 or G711FB. In a FAX scenario, when the codec policy adds either T.38 or G711FB, a new m= line is added to the SDP. When adding T.38, the new m= line specifies the T.38 codec. When adding G711FB, the new m= line specifies PCMU (or alternatively PCMA).

Once added, m= lines can not be deleted in the context of a call. The Oracle Enterprise Session Border Controller maintain all m= lines between itself and an endpoint throughout the course of call. All m= lines not in use can be disabled by setting their receive port to 0, but they can not be removed from the SDP.

Defining G711FB

G711 Fall Back (G711FB) is an internal codec that encompasses PCMU and PCMA for carrying fax information FAXable codecs. The G711FB codec must be configured either way for when the Oracle Enterprise Session Border Controller inserts a FAXable codec in SDP. G711FB is only used for FAX transcoding scenarios.

To define G711 FB, create a media profile configuration element named g711fb and set the payload-type to 0 or 8.

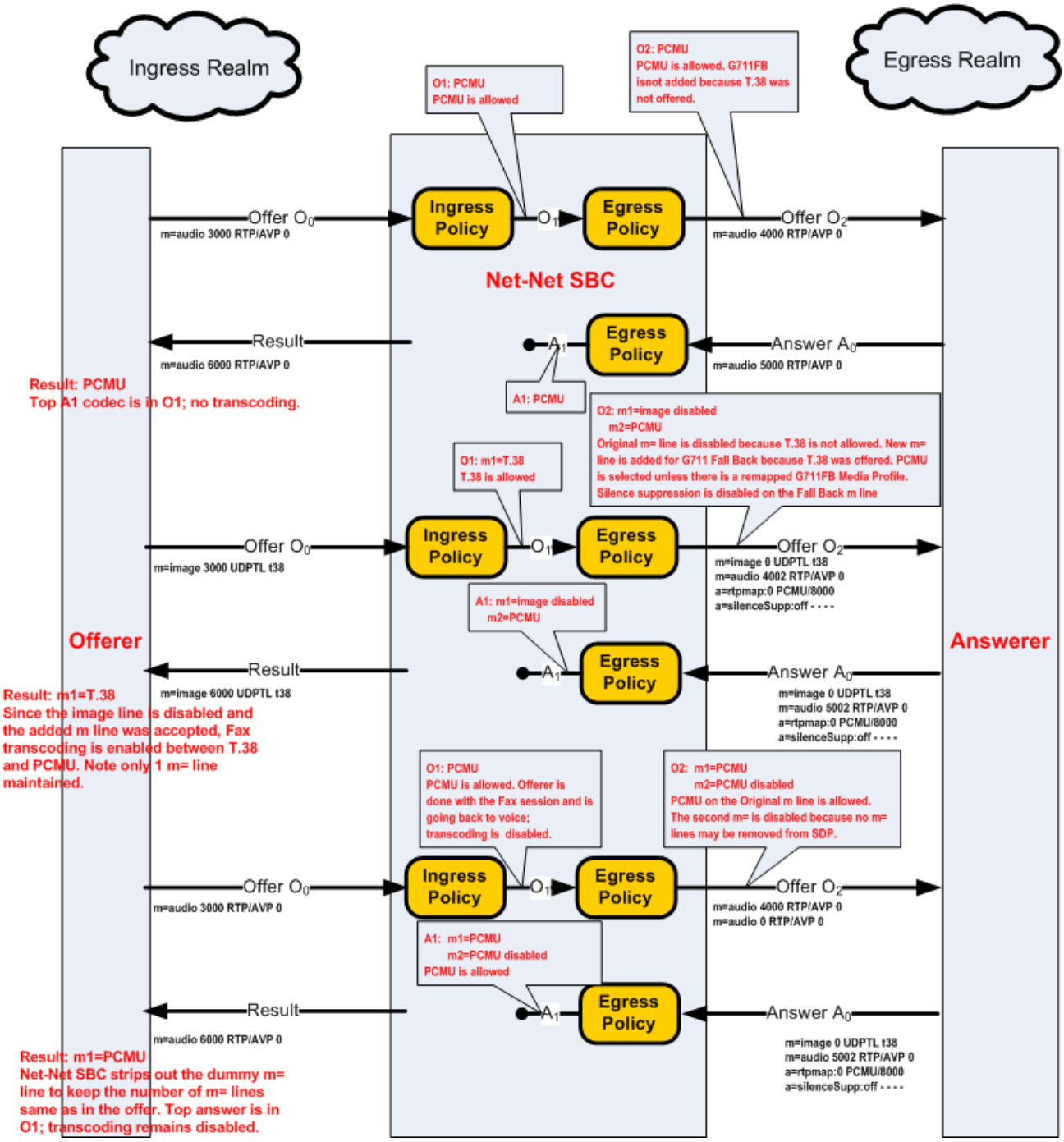
Codec (supported bit rates)	RTP Payload Type	Default Ptime (ms)	Supported Ptime (ms)
T.38	N/A	30	10, 20, 30
G711FB (64 kbps)	0, 8	30	10, 20, 30

FAX Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	T.38:no
add-codecs-on-egress		add-codecs-on-egress	G711FB
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

In the following diagram, there are three offer-answer exchanges. Initially a PCMU-to-PCMU session is negotiated. Next, a T.38 to PCMU session is negotiated. Finally, the session reverts to non-transcoded PCMU to PCMU state.



FAX Scenario 2

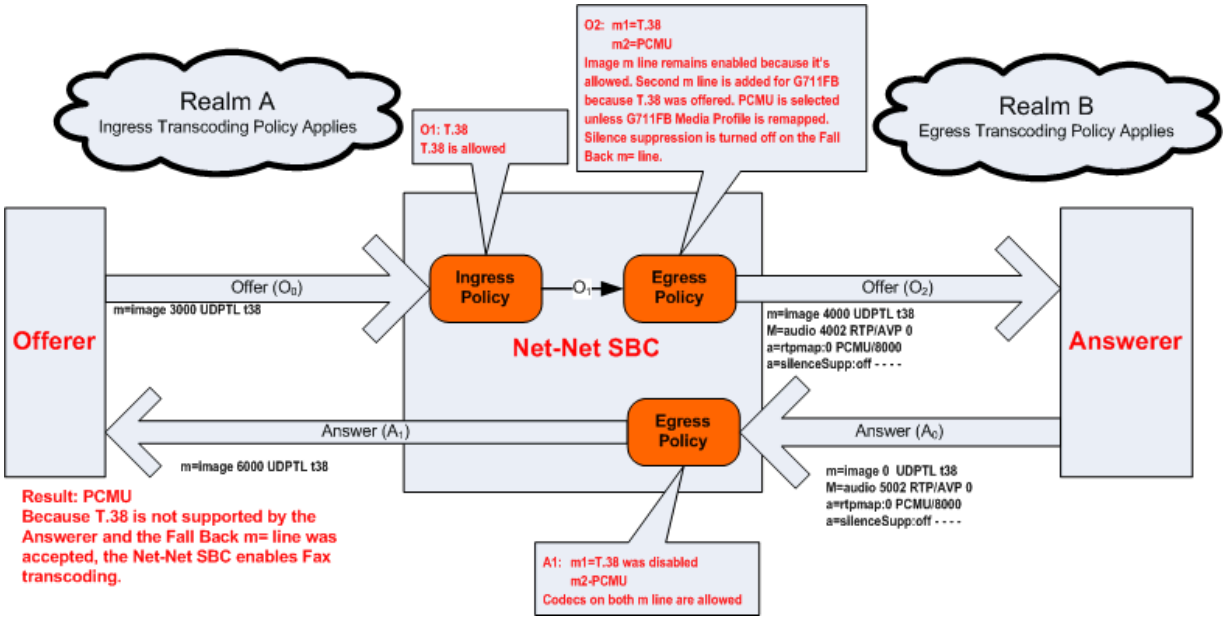
The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	G711FB
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled

Ingress Policy		Egress Policy	
packetization-time		packetization-time	

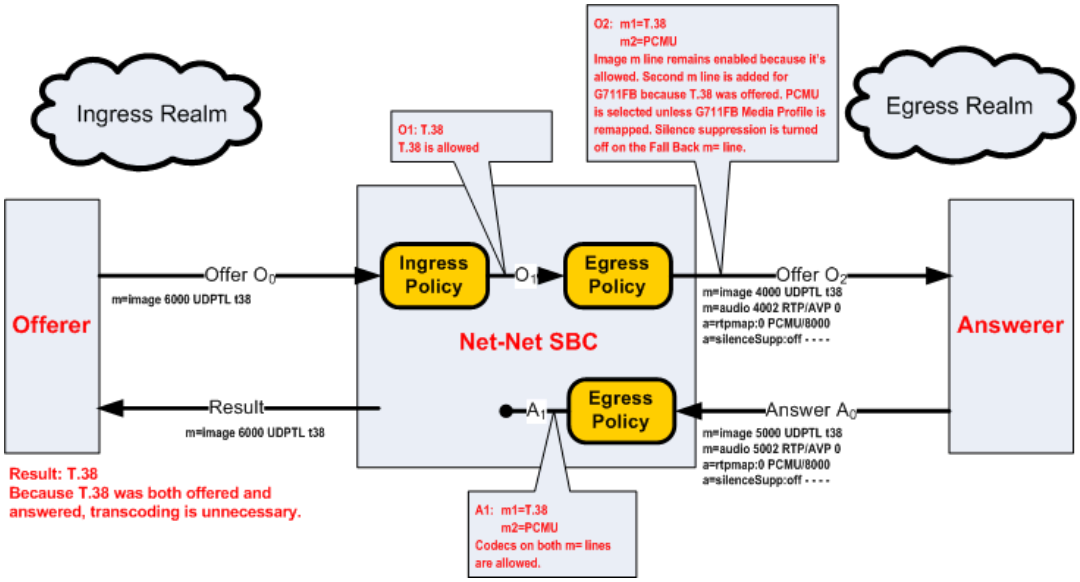
1. In the following diagram, T.38 is offered to the Oracle Enterprise Session Border Controller. A second m= line was added to O1 that included a G711FB codec (PCMU).

   The SDP answer agreed to PCMU, but disabled T.38. When the Oracle Enterprise Session Border Controller forwarded the SDP in A1 to the answerer, it stripped the second m= line. Because A1 rejects T.38 m= line, but accepts the PCMU m= line, FAX transcoding is enabled.



2. In the following diagram, T.38 is offered to the Oracle Enterprise Session Border Controller. A second m= line was added to O1 that included a G711FB codec (PCMU).

   The SDP answer agreed to PCMU and T.38. Because both O1 and A1 support T.38, the call proceeds without transcoding.



FAX Scenario 3

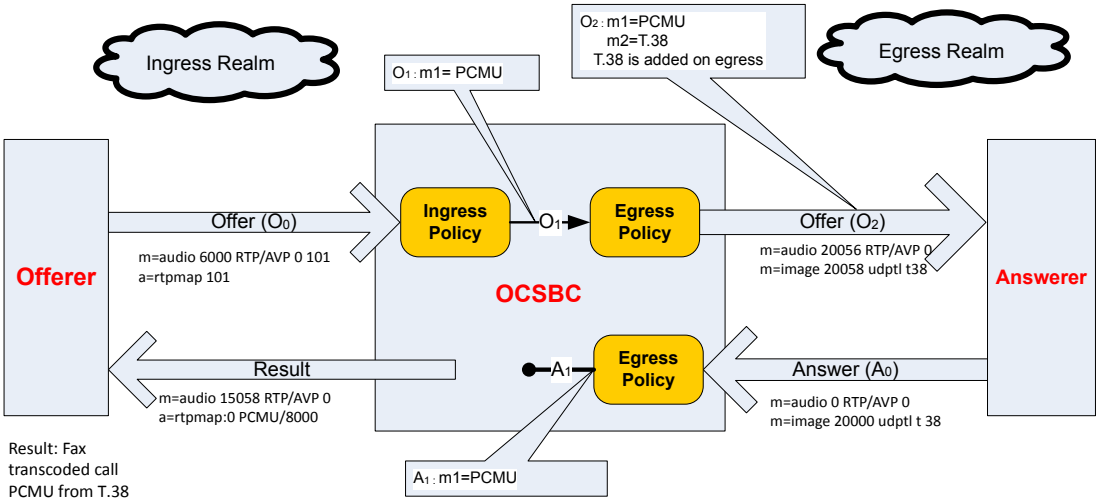
The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	PCMU, G729 ,T.38
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, PCMU and telephone-event codecs are received by Oracle Enterprise Session Border Controller .The egress codec policy has PCMU, G729 and T.38 **add-codecs-on-egress**.

   Since there is a faxable codec in the SDP offer and T.38 in **add-on-egress**, the non-faxable codec G729 is stripped and T.38 is added to egress offer.

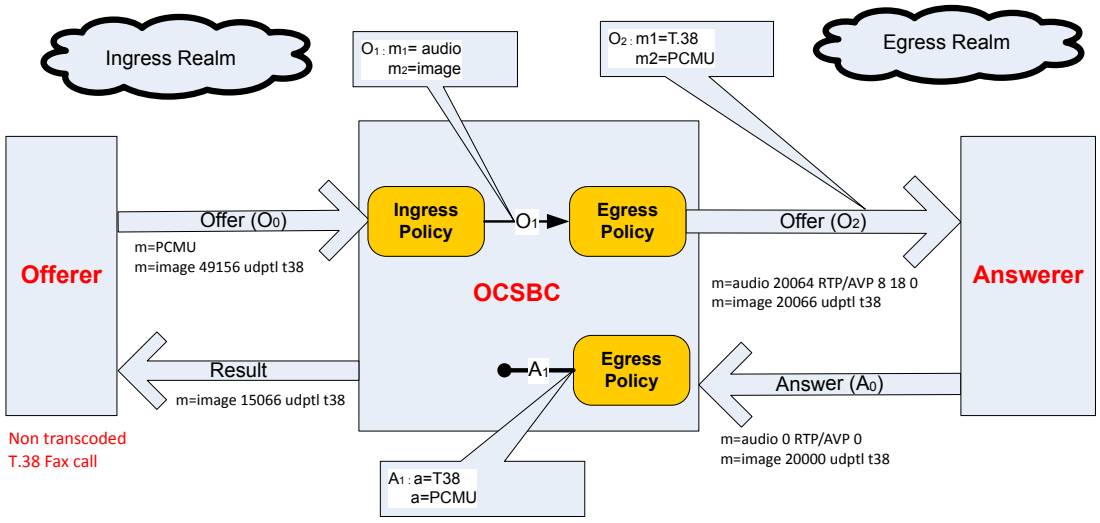
   In the A0 answer the audio m-line is zero indicating disabled and the port for the image m-line is non-zero (20000) so the called party has selected T.38. Hence PCMU in realm A is transcoded to T.38 in realm B.



2. In the following diagram, PCMU and T38 are received by the Oracle Enterprise Session Border Controller. The egress codec policy has PCMU, G729, and T.38 **add-codecs-on-egress**.

   T.38 is present in the O0 offer, and therefore will not be explicitly added by the egress codec policy to the O2 Offer. The logic to remove non-faxable codecs is only invoked when T.38 is added by the **add-codecs-on-egress** parameter. In this example, T.38 is not added since is already present in SDP. With PCMU and T.38 received, G729 as a non-faxable codec is not removed. Therefore, PCMU, T.38, and G729 are all present in the O2 SDP offer sent to the answerer.

   In the A0 answer, the audio m-line port is 0 indicating that audio is disabled, and the port for the image m-line is nonzero (20000), thus the called party has selected T.38. In the A1 answer the OCSBC send the audio m-line port as 0 indicating that audio is disabled, and the port for the image m-line is nonzero (20000). This set-up results in a non-transcoded T.38 -OCSBC -T.38 fax call.



Transrating

The Oracle Enterprise Session Border Controller can transrate media as it exits the Oracle Enterprise Session Border Controller into the network. Transrating is also known as forced packetization time (ptime), and is used to enforce a configured ptime within a realm. Transrating is often desirable when devices in a realm can only accept media with a specific ptime, or to optimize bandwidth.

If this feature is configured, the media portion of a call is transrated regardless of which codecs are ultimately chosen for each realm as long as they are transcodable. This allows realms that have devices that can only use a single packetization interval to interwork with devices that may or may not have the same packetization capabilities.

You must enable force-ptime in the egress codec policy and then specify the packetization time to force. When force ptime is enabled, it implicitly masks all codecs not of the specified packetization time that are listed in that codec policy's allow codecs and add codecs on egress parameters. For example, if force ptime is enabled with a packetization time of 20 ms, then no G723 codecs (which are only available at 30, 60, and 90 ms) may be active via codec policy in that realm.

Transrating occurs when forced-ptime is enabled and the offered and answered ptimes do not match and the top non-Signaling codec of A1 and top non Signaling codec of O1 are Transcodable.

Note: Answered ptime A1 does not have to be equal to the ptime inserted into the outgoing offer O2, it just has to be different than the offer the Oracle Enterprise Session Border Controller received (O1).

Please refer to *Transcodable Codecs* for current list of ptimes per codec supported by the DSPs.

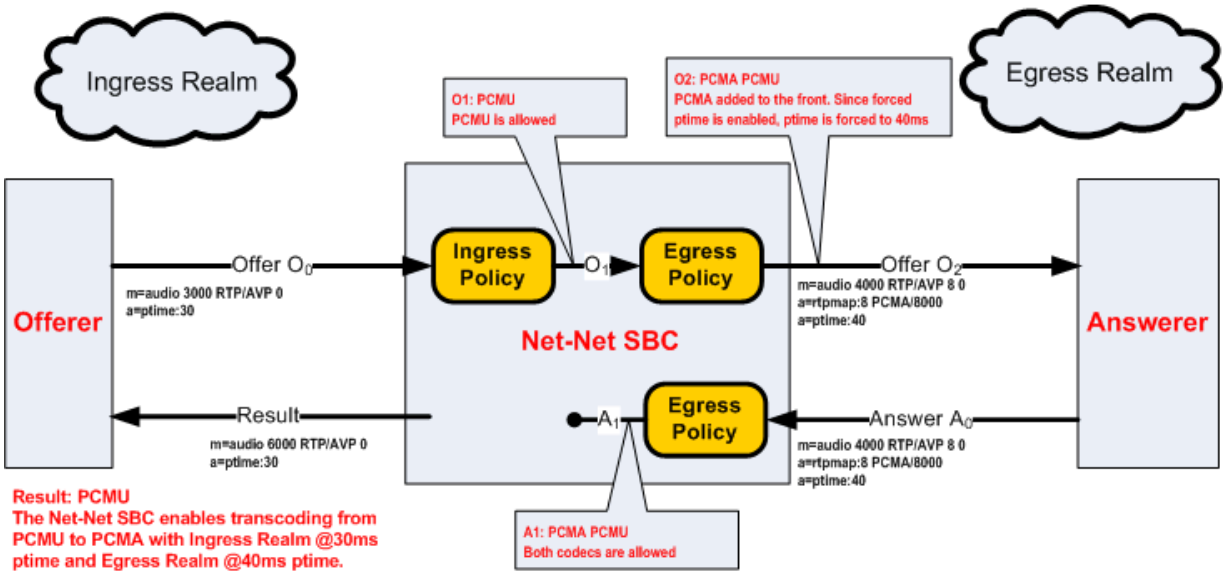
Transrating Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	PCMA
order-codecs	G723 *	order-codecs	
force-ptime	disabled	force-ptime	enabled
packetization-time		packetization-time	40

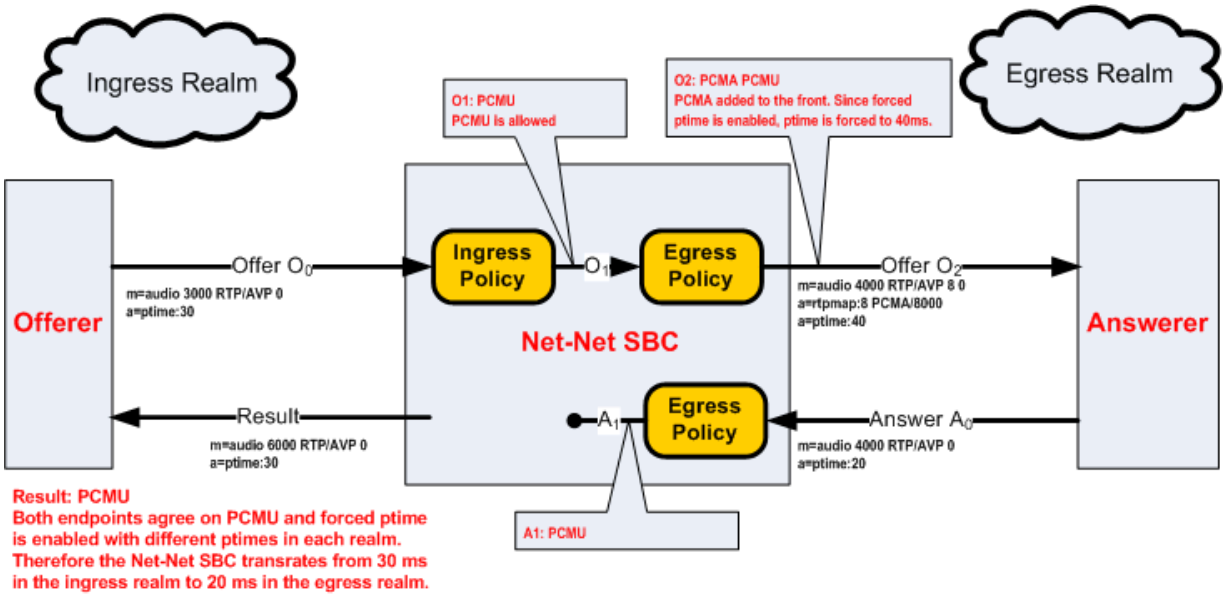
1. In the following diagram, PCMU is offered in the ingress realm with 30ms ptime, and the egress realm is forced to use 40ms ptime. PCMA is added as the top codec for the egress realm.

The Oracle Enterprise Session Border Controller enables transcoding between the ingress realm (PCMU) and the egress realm (PCMA) and the ptimes as negotiated are also maintained.



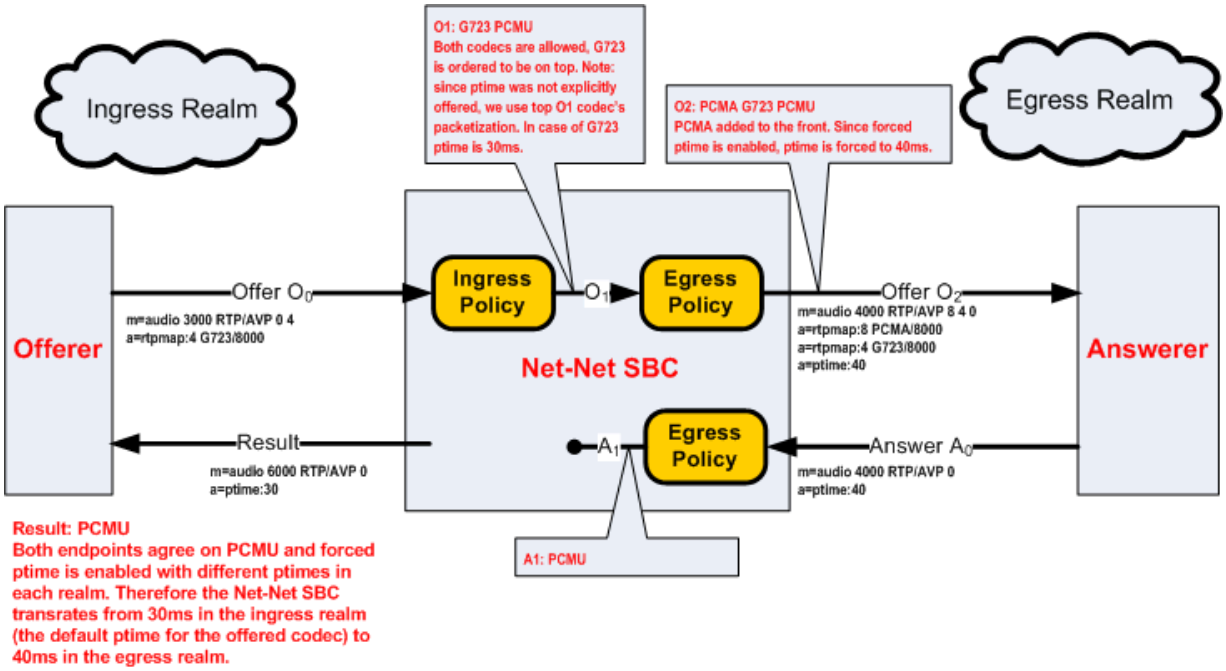
2. In the following diagram, PCMU is offered in the ingress realm with a ptime of 30ms, and forced to 40 ms in the egress realm by policy.

   The answerer chooses to use PCMU with a 20 ms ptime. Thus the call is not transcoded, but it is transrated from 30ms in the ingress realm to 20ms in the egress realm.



3. In the following diagram, PCMU and G723 are offered in Realm A. The top codec's ptime (30ms) is implied as the one for the ingress realm. The Oracle Enterprise Session Border Controller adds PCMA to the SDP offer with a 40ms ptime.

   The answerer chooses to use PCMU with a 40 ms ptime. Thus the call is transrated from 30ms in the ingress realm to 40ms in the egress realm.



Default Media Profiles

The Oracle Enterprise Session Border Controller contains a set of default media profiles that define characteristics of well-known IANA codecs. You can not view the default media profiles' configurations, but you can override them by configuring identically-named media profile configuration elements.

Transcodable codecs are a subset of the default media profiles which the Oracle Enterprise Session Border Controller can transcode between.

Transcodable Codecs

The following list shows the transcodable codecs which the Oracle Enterprise Session Border Controller can add to SDP. These codecs all reflect default media profiles for their given names. Enter codecs in the configuration exactly as below.

- PCMU
- PCMA
- G729
- G729A
- iLBC
- telephone-event
- T.38
- G711FB
- G726
- G726-16
- G726-24
- G726-32
- G726-40
- G722
- G723
- GSM
- AMR

- AMR-WB
- EVRC0
- EVRC
- EVRC1
- EVRCB0
- EVRCB
- EVRCB1

When creating an override media profile from the previously listed codec, case is ignored. Also, GSM is GSM-FR.

Preferred Default Payload Type

When the Oracle Enterprise Session Border Controller adds a codec with a dynamic payload type to SDP, it uses the lowest unused payload number. You can configure a preferred payload type for a dynamic codec by creating an override media profile. This makes the Oracle Enterprise Session Border Controller use your preferred payload type for insertion into SDP. If you configure a dynamic codec to use a preferred payload type, and that payload type is already in use, the codec will still be inserted into SDP, but with the first available dynamic payload type.

For example, you create a media profile for telephone-event with a payload type of 101. If telephone-event is added to SDP, and payload type 101 is already in use in the SDP, the Oracle Enterprise Session Border Controller will use the first available payload type in the 96-127 range when adding telephone-event.

Redefining Codec Packetization Time

You can configure a media profile with a packetization time (ptime) that overrides the codec's default ptime. Transcoding functions look up and use default ptimes when not specified in offered or answered SDP. Default ptime for most audio codecs is 20ms; some however are 30ms. See the *Transcodable Codecs* list for default values.

To change the default ptime for a codec, you must create a media profile that overwrites the default ptime parameter with your new packetization time. When SDP is received with no 'a= ptime' attribute or when adding the codec to egress SDP, the newly configured ptime is used.

New default ptime for a media profile is entered by typing "ptime=<x>" in the parameters parameter, where <x> is the new default packetization time.

mptime Support for Packet Cable

The SDP specification lacks the ability to specify unique packetization times per codec when more than one codec is listed in an m= line. The ptime attribute is not related to a specific codec but to the entire m= line. When multiple codecs appear on a single m= line, the PacketCable mptime attribute can specify different packetization times for each codec.

The Oracle Enterprise Session Border Controller adheres to PKT-SP-NCS1.5-I01-050128 and PKT-SP-EC-MGCP-I06-021127 for processing and generating mptime. The mptime line uses an integer to indicate the packetization time for each corresponding codec in the m= line. The dash character, "-", on an mptime line is used for non-packetized codecs, such as CN or telephone-event.

If the Oracle Enterprise Session Border Controller receives an invalid mptime, it is ignored and removed. If a valid mptime is received in the incoming SDP, its values will be used for packetization times of each corresponding codec and a valid mptime line will be sent in the outgoing SDP.

Valid:

```
m=audio 10000 RTP/AVP 0 96 8
a=mptime:20 - 30
a=rtpmap:96 telephone-event/8000
```

Valid: 'ptime' attribute is ignored

```
m=audio 10000 RTP/AVP 0 8
a=mptime:20 30
a=ptime:30
```

Invalid: dash cannot be first mptime value

```
m=audio 10000 RTP/AVP 96 0
a=mptime: - 20
```

When Oracle Enterprise Session Border Controller includes an mptime in an outgoing SDP, it will also always add a ptime attribute with the value of the most preferred codec. This is done to increase the interoperability with devices that do not support mptime.

AMR-NB and AMR-WB Specifications

The Oracle Enterprise Session Border Controller supports Adaptive Multi-Rate Narrow Band & Wide Band codecs. All configurations of this codec, as indicated by SDP, are transcodable except when the following SDP parameters are enabled:

- robust-sorting
- interleaving

When AMR is configured in a codec policy's add-codecs-on-egress parameter, it is forwarded from the Oracle Enterprise Session Border Controller with the following default settings:

- 12.2 kbps (AMR-NB)
- 23.85 kbps (AMR-WB)
- RTP/IF1 format
- No redundant packets
- bandwidth efficient default payload
- No CRC frame
- 20ms default ptime

Note: AMR and AMR-WB each require a separate license.

Configuring Transcoding

Codec Policy Configuration

Transcoding is configured by creating codec policies and referencing them from a realm configuration.

ACLI Configuration Instructions

The parameters that you can configure are name, allow-codecs, add-codecs-on-egress, order-codecs, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Note: A single codec policy can be reused for any number of realms.

To access the configuration parameters for codec policies:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type codec-policy and press Enter.

   ```
   ACMEPACKET(media-manager)# codec-policy
   ```

   From this point, you can start configuring your codec policy.

Naming Codec Policies

The codec policy's name is important not only because it uniquely identifies the policy, but because it is the name you will enter into your realm configuration's codec-policy parameter. It is important to apply the correct policy to the appropriate realm.

To set the codec policy's name:

name—Set the name for this codec policy, and note it for future reference when you apply codec policies to realms. This parameter is required, and has no default.

```
ACMEPACKET(codec-policy)# name private
```

Removing Allowing and Adding Codecs

The Oracle Enterprise Session Border Controller removes and allows codecs using the allow-codecs parameter. Refer to the *Codec Policy Definition* section of this chapter for configuration information.

- allow-codecs—The allow-codecs parameter takes a list of codecs that you want to pass through the Oracle Enterprise Session Border Controller and can explicitly allow them to remain in the SDP for the next step; codecs not matching the items on this list are removed. This parameter is required.
- add-codecs-on-egress—The add-codecs-on-egress parameter sets the codecs that the Oracle Enterprise Session Border Controller adds to an offer if that codec is not already there. This parameter applies only to the egress policy.

  For allow-codecs, order-codecs, and add codecs to codec policies:

You can configure and edit these two transcoding parameters as ACLI lists, meaning that there are add and delete commands associated with each. You type the name of the parameter, choose the operation you want to perform on the list (adding or deleting), and then specify the data that you want to add or remove.

The examples in the procedure that follows show you how to add to the lists you are configuring. To remove items from the allow-codecs list, simply replace the add command you see in these example with delete and the items you want to remove.

If you want to overwrite previous values, you can enter the command, a Space and the items in the list enclosed in quotes (“).

1. allow-codecs—Enter a list of codecs that are allowed to pass through the Oracle Enterprise Session Border Controller. Use the syntax in the *Transcodable Codecs* section of this chapter. To allow all codecs, enter an asterisk (*).

   ```
   ACMEPACKET(codec-policy)# allow-codecs *
   ```

   When multiple items are added, enclose them in quotes. For example:

   ```
   ACMEPACKET(codec-policy)# allow-codecs G729 G711 AMR
   ```

2. add-codecs-on-egress—Enter the codecs that you want added to the SDP offer for the egress codec policy. If you leave this parameter blank, then the Oracle Enterprise Session Border Controller will not add codecs to the SDP answer. This parameter cannot be wildcarded; other possible value are listed in the *Transcodable Codecs* section of this chapter.

   ```
   ACMEPACKET(codec-policy)# add-codecs-on-egress G729
   ```

   If you need to modify the list of configured codecs, you must enter the complete list at once.

Ordering Codecs

Codec policy can specify the order that codecs appear in the SDP offer or answer. Refer to the *Codec Policy Definition* section of this chapter for configuration information.

To configure an order which codecs appear in the offer:

order-codecs—Enter the order in which you want codecs to appear in the SDP offer or answer. You can enter them in any of the ways described in the preceding explanation.

```
ACMEPACKET(codec-policy)# order-codecs G711 * G729
```

Transrating Configuration

The following procedure explains how to configure transrating for a codec policy. This codec policy must be applied as an egress codec policy.

To configure forced ptime for a codec policy:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type network-parameters and press Enter.

```
ACMEPACKET(media-manager)# codec-policy
ACMEPACKET(codec-policy)#
```

4. If you are adding support for this feature to a pre-existing configuration, then you must select the specific configuration instance, using the ACLI select command.

```
ACMEPACKET(codec-policy)# select 1
```

   You can now configure forced ptime.

5. force-ptime—Set this parameter to enabled to enable forced ptime for this codec policy.
6. packetization-time—Enter the ptime in ms to use in the realm where this codec policy is active. Valid values are 10, 20, 30, 40, 50, 60, 70, 80, 90 ms. The default value is 20 ms.
7. Save your work using the ACLI done command.

Applying a Codec Policy to a Realm

Once you have configured a codec policy, you apply it to a realm by configuration name.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
```

4. codec-policy—Enter the name of the codec policy that you want to apply to this realm. This value is the same as the one you entered in the name parameter for the codec policy you want to use for this realm. There is no default for this parameter.

```
ACMEPACKET(realm-config)# codec-policy private
```

Media Profile Configuration

Media profiles must be created and then defined when you want to override the Oracle Enterprise Session Border Controller's default media profiles.

ACLI Configuration Instructions and Examples

The parameters that you can configure are name, allow-codecs, add-codecs-on-egress, order-codecs, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Creating User-Defined Ptime per Codec

To change the Oracle Enterprise Session Border Controller's default ptime for a specific codec, you must create a media profile configuration element. In the parameter parameter, you set the ptime to the value of your choosing.

Note: The frames-per-packet parameter in the media profile configuration element is NOT used for setting a user defined ptime for that codec.

To configure a new ptime value for a codec:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

   If you are adding ptime to a pre-existing media profile, then you must select (using the ACLI select command) the configuration that you want to edit. If you are adding ptime to an undefined media profile, you must create it first.

4. name—Type the name of the codec for which you are creating a new default ptime.

```
ACMEPACKET(media-profile)# name pcmu
```

5. payload-type—Enter the well-known payload type for this codec.

```
ACMEPACKET(media-profile)# payload-type 0
```

6. parameter—Set the ptime by typing parameter, a Space, ptime=, the new ptime value. Then press Enter. For example:

```
ACMEPACKET(media-profile)# parameter ptime=40
```

7. Save your work using the ACLI done command.

Media Type Subnames

You can define multiple versions of a media profile for a single codec by using the subnames feature. You can then reference the new media profile by a combination of the media profile name and media profile subname.

Some media types are not unique per just their value in an SDP m= line, they must be uniquely identified by looking at additional SDP parameters. For example, you can define a media profile for G729, when only the parameter and value annexb=yes is present in the SDP. By creating a media profile + subname that defines both a media type and parameter, you can perform various operations on G729 only when annexb=yes is encountered.

Some applications of media type subnames are:

- maintaining different versions of the same codec with different bandwidth ceilings
- maintaining different versions of the same codec with different ptimes
- grouping codecs by using customer as a subname
- grouping codecs by using realm as a subname

SDP Parameter Matching

This feature matches parameters in the a=fmtp, codec-specific SDP a= line. It does not try to match a global m= line attribute like a=ptime.

Using Subnames with Codec Policies

Media profiles are defined and referenced in the ACLI by a name and subname in the following format

```
<name>::<subname>
```

If no subname has been created for a media profile, you may continue using the media profile name without any subname specifier.

For example, to remove a media profile and subname configured as PCMU::customer1 from all SDP entering the egress realm, you would configure the codec policy allow-codecs parameter as follows:

```
allow-codecs PCMU::customer1:no
```

Subname Syntax and Wildcarding

You can wildcard one or both portions (name and subname) of a media type and subname pair:

- When you wildcard the name portion of the value, you can provide a specific subname that the Oracle Enterprise Session Border Controller uses to find matching media profiles.
- When you wildcard the subname portion of the value, you can provide a specific name that the Oracle Enterprise Session Border Controller uses to find matching media profiles.

The following table defines and explains subname wildcarding and syntax:

Syntax	Example Value	Description
<name>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <name>::*.
<name>::	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<name>::*	PCMU::*	Matches any and all media profiles with the name value configured as PCMU with any and all subname configured.
<name>::<subname>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
*	*	Matches anything, but does not have to be a defined media profile.
::	*::*	Matches any and all media profiles, but requires the presence of media profile configurations.
*::<subname>	*::64k	Matches all media profiles with this subname. You might have a group of media profiles with different names, but the same subname value.
*::	*::	Matches any media profiles with an empty subname parameter.
::	::	Invalid
::*	::*	Invalid

Wildcarding add-codecs-on-egress

It is important to note that you may not configure add-codecs-on-egress with a wildcarded subname in a codec policy. You may only add a specific instance of a media type.

Valid:

```
add-codecs-on-egress PCMU
add-codecs-on-egress PCMU::customer1
```

Invalid:

```
add-codecs-on-egress PCMU::*
```

Media Type and Subname Configuration

To use media type subnames with a codec policy, you must first configure a media profile and subname. Then you can configure a codec policy with a media type and subname pair for your application

To use configure a media type and subname:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type media-profile and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. name—Type the name of the codec for which you are creating a new default ptime.

```
ACMEPACKET(media-profile)# name g729
```

5. subname—Enter a description for the use of this subname

```
ACMEPACKET(media-profile)# subname annexb-yes
```

   You may now configure this subname's unique attributes. PCMU is created with ptime of 30 in this example.

6. parameters—Set the ptime by typing parameter, a Space, ptime=, the new ptime value. Then press Enter. For example:

```
ACMEPACKET(media-profile)# parameter annexb=yes
```

   Note: Remember to configure all additional, required media profile parameters, or they will inherit default values.

7. Save your work using the ACLI done command.

Codec Policy Configuration with a Media Type with a Subname

To configure a codec policy with a media type with subname:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type codec-policy and press Enter.

```
ACMEPACKET(media-manager)# codec-policy
```

4. Use the ACLI select command to select a codec policy.

```
ACMEPACKET(codec-policy)# select 1
```

   You may now enter a media profile with subname to any parameter in the codec policy that accepts a media profile.

5. allow-codecs—Enter a list of codecs that this codec policy allows or denies from passing through the Oracle Enterprise Session Border Controller. To allow all codecs, enter an asterisk (*).

```
ACMEPACKETcodec-policy# allow-codecs g729::annexb-yes:no
```

6. Save and activate your configuration.

Codec and Conditional Codec Policies for SIP

The Oracle Enterprise Session Border Controller has the ability to add, strip, and reorder codecs for SIP sessions. This builds on the Oracle Enterprise Session Border Controller's pre-existing abilities to route by codec and re-order one codec in an SDP offer by allowing you to configure the order of multiple codecs and to remove specific codecs within the media descriptions in SDP offers.

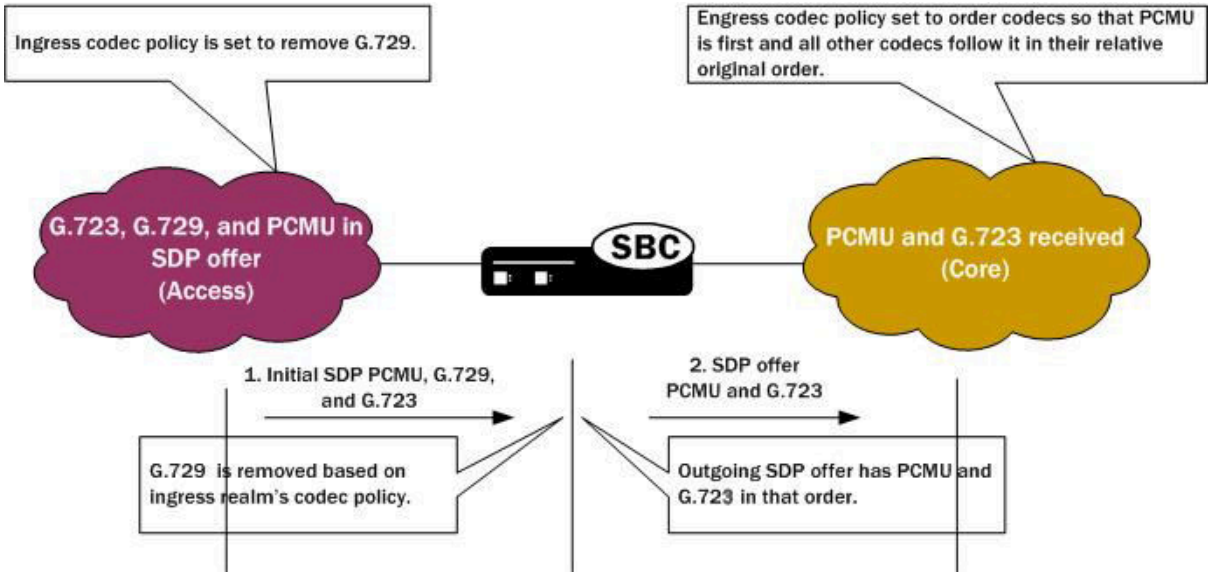
You can enable the Oracle Enterprise Session Border Controller to perform these operations on SDP offers by configuring codec policies. Codec policies are sets of rules that specify the manipulations to be performed on SDP offers. They are applied on an ingress and egress basis using the realm and session agent configurations.

Oracle Enterprise Session Border Controller supports three types of codec policies:

- Ingress policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for incoming traffic
- Egress policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for traffic leaving the Oracle Enterprise Session Border Controller
- Conditional policy—Codec policy that the Oracle Enterprise Session Border Controller applies to the SDP offer for traffic leaving the Oracle Enterprise Session Border Controller. A conditional policy differs from an egress policy in providing the capability to perform standard codec manipulations (add, strip and re-order) dynamically, based on the codec list and associated parameters contained in the original SDP offer. Refer to *Conditional Codec Policies* for specific details regarding the use and construction of these policies.

The Oracle Enterprise Session Border Controller applies codec policies during the offer phase of media format negotiation. If codec manipulation is enabled, then the Oracle Enterprise Session Border Controller performs the modification according to the specific policy and forwards on the traffic.

For example, when the Oracle Enterprise Session Border Controller receives a SIP INVITE with SDP, it refers to the realm through which the INVITE arrived and performs any manipulations specified by an ingress codec policy that may have been assigned to the ingress realm. With the media description possibly changed according to the ingress codec policy, the Oracle Enterprise Session Border Controller passes the SDP offer to the outgoing realm so that the an egress codec policy can be applied. Note that the SDP to be evaluated by the egress codec policy may match the original SDP, or it may have been changed during transit through the ingress realm. After applying the egress coded policy, the Oracle Enterprise Session Border Controller forwards the INVITE.



Since the offer-answer exchange can occur at different stages of SIP messaging, the assigned ingress and egress roles follow the media direction rather than the signaling direction. It might be, for example, that the offer is in an OK that the Oracle Enterprise Session Border Controller modifies.

You can apply codec policies to realms and to session agents; codec policies configured in session agents take precedence over those applied to realms. However, it is not required that there be both an ingress and an egress policy either for realms or for session agents. If either one is unspecified, then no modifications take place on that side. If neither ingress nor egress policies specified, SDP offers are forwarded as received.

Relationship to Media Profiles

For each codec that you specify in a codec policy, there must be a corresponding media profile configuration on the Oracle Enterprise Session Border Controller. You configure media profiles in the ACLI via the session-router path. In them, you can specify codec type, transport protocol, required bandwidth, and a number of constraints.

Manipulation Modes

You can configure a codec policy to perform several different kinds of manipulations:

- Allow—List of codecs that are allowed for a certain codec policy; if a codec does not appear on this list, then the Oracle Enterprise Session Border Controller removes it. You can wildcard this list with an asterisk (*) so that all codecs are allowed. Further, you can create exceptions to a wildcarded allow list.
  - You make an exception to the wildcarded list of codecs by entering the codec(s) that are not allowed with a no attribute. This tells the Oracle Enterprise Session Border Controller to allow all codecs except the one(s) you specify.

    ```
    ACMEPACKET(codec-policy)# allow-codecs (* PCMA:no)
    ```
  - You can also create exceptions to allow lists such that audio or video codecs are removed. However, when the allow list specifies the removal of all audio codecs and an INVITE arrives at the Oracle Enterprise Session Border Controller with only audio codecs, the Oracle Enterprise Session Border Controller behaves in accordance with RFC 3264. This means that the resulting SDP will contain one attribute line, with the media port for the media line set to 0. The terminating side will need to supply new SDP in its reply because the result of the manipulation is the same as an INVITE with no body.

    ```
    ACMEPACKET(codec-policy)# allow-codecs (* audio:no)
    ```
- Order—List of the codecs where you specify their preferred order in the outgoing media offer. The Oracle Enterprise Session Border Controller arranges matching codecs according to the rule you set, and any remaining ones are added to the list in the same relative order they took in the incoming media offer. If your list specifies a codec that is not present, then the ordering proceeds as specified but skips the missing codec.

  You can use an asterisk (*) as a wildcard in this list, too. The placement of the asterisk is key, as you can see in the following examples:

  - For an order rule set this way

    ```
    ACMEPACKET(codec-policy)# order (A B C *)
    ```

    codecs A, B, and C will be placed at the front of the codec list in the order specified; all other codecs in the offer will follow A, B, and C in the same relative order they had in the original SDP offer.
  - For an order rule set this way:

    ```
    ACMEPACKET(codec-policy)# order (* A B C)
    ```

    codecs A, B, and C will be placed at the end of the codec list in the order specified; all other codecs in the offer will come before A, B, and C in the same relative order they had in the original SDP offer.
  - For an order rule set this way

    ```
    ACMEPACKET(codec-policy)# order (A * B C)
    ```

    codec A will be placed at the beginning of the codec list, to be followed by all other codecs in the offer in the same relative order they had in the original SDP offer, and then B and C will end the list.
- Force—An attribute you can use in the allow list with one codec to specify that all other codecs should be stripped from the outgoing offer. You can specify multiple forced codecs in your rules.
  - If you set multiple codecs in the allow list and one of them is forced, then the outgoing offer will contain the forced codec.
  - If you set multiple codecs in the allow list and the one that is forced is not present in the offer, then the Oracle Enterprise Session Border Controller will select a non-forced codec for the outgoing offer.

    ```
    ACMEPACKET(codec-policy)# allow (PCMU G729:force)
    ```

    You cannot use the force attribute with a wildcarded allow list.

- No—An attribute that allows you to strip specified codecs or codec types from a wildcarded allow list.

```
ACMEPACKET(codec-policy)# allow (* PCMA:no)
```

In-Realm Codec Manipulation

In addition to being able to apply codec policies in realms, the realm configuration supports a setting for determining whether codec manipulation should be applied to sessions between endpoints in the same realm.

In-realm codec manipulation can be used for simple call flows that traverse two realms. If the originating and terminating realms are the same, the Oracle Enterprise Session Border Controller checks to see if you have enabled this capability. If you have enabled it, then the Oracle Enterprise Session Border Controller performs the specified manipulations. If this capability is not enabled, or if the realm's media management in realm (mm-in-realm) setting is disabled, then the Oracle Enterprise Session Border Controller does not perform codec manipulations.

For more complex calls scenarios that involve call agent or reinitiation of a call back to the same realm, the Oracle Enterprise Session Border Controller does not perform in-realm codec manipulation.

Conditional Codec Policies

A codec policy performs actions conditionally when any of its parameters includes a conditional value. A conditional value includes a target codec paired with a requirement for executing the action. The Oracle Enterprise Session Border Controller manipulates SDP according to this value pair if the ingress SDP or a previous manipulation to the SDP meets the condition's criteria. The user configures this "conditional manipulation" by extending upon the syntax of three core parameters in the codec-policy configuration element, including:

- allow-codecs,
- add-codecs-on-egress, and
- order-codecs.

The system establishes conditions on a codec policy as a sequence of allowing, adding, and re-ordering. Each step in this sequence can occur with or without conditions. Allowing is required. Any applied policy, whether or not it is conditional, without an allow blocks all traffic. Allow all, using the wildcard asterisk character is a typical setting. Adding only applies to egress policies.

To establish conditions, the user configure the parameter with pairs that consist of target codecs followed by the condition(s) that trigger the action. Each policy parameter can include one or more of these pairs. When configuring a parameter with multiple values, the user encloses them within parenthesis, whether or not there are conditions.

The system processes all policies serially, regardless of whether any includes a condition. Specifically, the system first determines which codecs to allow, then which to add (none on ingress), then the codec order. The system also processes parameter values serially. The user, therefore, must configure all parameter values based on what may have been changed previously. This is particularly important when using both ingress and egress codec policies. Consistent with this serial concept, egress policies operate on what the system presents to them, which includes the results of any ingress policies.

Note that the use of conditional **order-codecs** is often done in conjunction with **add-codecs-on-egress** to define the location of user-added codecs to the list presented at egress. When **order-codecs** is not used, the system places all added codecs at the beginning of the list in the order they were added. It is often desirable to place an added codec in a different position, using **order-codecs**.

Conditional Codec Lists

Conditional codec policies are constructed using existing ACLI configuration commands — add-codecs-on-egress and allow-codecs — in conjunction with new keywords and operators. Conditions are defined by a continuous character string (no spaces allowed) that starts with an :(character sequence and is terminated by a closing parenthesis,). For example,

```
ACMEPACKET(codec-policy)# add-codecs-on-egress PCUM:(PCMA)
```

which can be interpreted as follows add PCMU if PCMA codec is in the SDP offer after ingress codec policy processing.

If PCMA is in the SDP offer after ingress codec policy processing, the add-codecs-on-egress ACLI command is treated as add-codecs-on-egress PCMU. If PCMA is not in the SDP offer after ingress codec policy processing, add-codecs-on-egress is treated as empty.

Both the conditioned codec and/or the condition itself can contain subnames. For example,

```
ACMEPACKET(codec-policy)# add-codecs-on-egress AMR::ONE:(AMR::TEST0)
```

which can be interpreted as follows add AMR::ONE if AMR::TEST0 codec is in the SDP offer after ingress codec policy processing.

Codecs contained in the condition can be wildcarded. For example,

```
ACMEPACKET(codec-policy)# add-codecs-on-egress AMR::ONE:(AMR::*)
```

which can be interpreted as follows add AMR::ONE if any AMR codec is in the offer after ingress codec policy processing.

Conditional Codec Operators

Three logical operators are available to construct conditional lists

the OR operator (|)

```
ACMEPACKET(codec-policy)# add-codecs-on-egress AMR::ONE:(AMR::*|PCMU)
```

which can be interpreted as — add AMR::ONE if any AMR:: codec is in the SDP offer after ingress codec policy processing, or if PCMU is in the SDP offer after ingress codec policy processing.

the AND operator (&)

```
ACMEPACKET(codec-policy)# add-codecs-on-egress AMR::ONE:(AMR::*&PCMU)
```

which can be interpreted as — add AMR::ONE if any AMR:: codec is in the SDP offer after ingress codec policy processing, and if PCMU is in the SDP offer after ingress codec policy processing.

the NOT operator (!)

```
ACMEPACKET(codec-policy)# add-codecs-on-egress AMR::ONE:(!AMR::*)
```

which can be interpreted as — “add AMR::ONE if no AMR:: codec is in the SDP offer after ingress codec policy processing.

Each operator applies only to the codec immediately following it. Operators are processed left to right until all conditions have been tested. The condition result is accumulated as each of the conditions is processed. For example:

```
ACMEPACKET(codec-policy)#add-codecs-on-egress AMR::ONE:
(!AMR::TEST1&AMR::TEST0|AMR::TEST2)
```

- Test SDP offer for AMR::TEST1 (a NOT operation).

  If AMR:TEST1 is NOT present, set accumulated result to TRUE.

  If AMR:TEST1 is present, set accumulated result to FALSE.
- Test SDP offer for AMR::TEST0 (an AND operation).

  If AMR is present, no change to accumulated result.

  If AMR is not present, set accumulated result to FALSE.
- Test SDP offer for AMR::TEST2 (an OR operation).

  If AMR is present, set accumulated result to TRUE.

  If AMR is not present, no change to accumulated result.

Multiple conditions can be concatenated; in this case, individual conditions are separated by SPACE characters and the ACLI command argument is bracketed with double quotation marks ...). For example:

```
ACMEPACKET(codec-policy)#add-codecs-on-egress (PCMU G729:(G726) G723:(PCMA))
```

- PCMU is unconditionally added to the egress codec list.

- Process the first condition — G729:(G726)

  If G726 is present, the result is TRUE; add G729 to the egress codec list.

  If G726 is not present, the result is FALSE; do not add G729 to the egress codec list.
- Process the second condition — G723:(PCMA)

  If PCMA is present, the result is TRUE; add G723 to the egress codec list.

  If PCMA is not present, the result is FALSE; do not add G723 to the egress codec list.

ACLI Instructions and Examples

This section gives instructions and examples for how to configure codec policies and then apply them to realms and session agents. It also shows you how to configure settings for in-realm codec manipulation.

Creating a Codec Policy

To create a codec policy:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type media-manager and press Enter to access the signaling-related configurations.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```
3. Type codec-policy and then press Enter.

   ```
   ACMEPACKET(media-manager)# codec-policy
   ACMEPACKET(codec-policy)#
   ```
4. name—Enter the unique name for the codec policy. This is the value you will use to refer to this codec policy when you apply it to realms or session agents. This parameter is required and is empty by default.
5. allow-codecs—Enter the list of media format types (codecs) to allow for this codec policy. In your entries, you can use the asterisk (*) as a wildcard, the force attribute, or the no attribute so that the allow list you enter directly reflects your configuration needs. Enclose entries of multiple values in parentheses (( )). For more information, refer to the *Manipulation Modes* section above.

   The codecs that you enter here must have corresponding media profile configurations.

   allow-codecs can be used to construct ingress, egress, or conditional codec policies. For details of conditional codec policies, refer to *Conditional Codec Policies*.
6. add-codecs-on-egress—Enter the codecs that the Oracle Enterprise Session Border Controller adds to an egress SDP offer if that codec is not already there. This parameter applies only to egress offers. For more information, refer to the *Manipulation Modes* section above.

   The codecs that you enter here must have corresponding media profile configurations.

   add-codecs-on-egress can be used to construct ingress, egress, or conditional codec policies. For details of conditional codec policies, refer to *Conditional Codec Policies*.
7. order-codecs—Enter the order in which you want codecs to appear in the outgoing SDP offer. Remember that you can use the asterisk (*) as a wildcard in different positions of the order to directly reflect your configuration needs. Enclose entries of multiple values in parentheses (( )). For more information, refer to the *Manipulation Modes* section above.

   The codecs that you enter here must have corresponding media profile configurations.
8. Save and activate your configuration.

   Your codec policy configuration will resemble the following example:

   ```
   codec-policy
           name            private
           allow-codecs    g723:no pcmu video:no
           order-codecs    pcmu *
   ```

Applying a Codec Policy to a Realm

Note that codec policies defined for session agents always take precedence over those defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. codec-policy—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

Applying a Codec Policy to a Session Agent

Note that codec policies that are defined for session agents always take precedence over those that are defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type session-agent and press Enter.

   ```
   ACMEPACKET(session-router)# session-agent
   ```

   If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the realm that you want to edit.

4. codec-policy—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

In-Realm Codec Manipulations

To enable in-realm codec manipulations:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ```

3. Type realm-config and press Enter.

   ```
   ACMEPACKET(media-manager)# realm-config
   ```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the realm that you want to edit.

4. codec-manip-in-realm—Enter the name of the codec policy that you want to apply to this realm. The default value is disabled. The valid values are:

   - enabled | disabled

5. Save and activate your configuration.

Transcoding Support for Asymmetric Dynamic Payload Types

Transcoding Support for Asymmetric Dynamic Payload Types supports the case when asymmetric payload types such that the RTP offered with one payload type and answered with another payload type will be acceptable to the Oracle Enterprise Session Border Controller when performing transcoding.

In certain network environments, MSC (Mobile Switching Center) equipment may require that originally offered (PT) payload type mappings be retained for the session duration, even if they are have been subsequently re-mapped as a result, for instance, of a RE-INVITE, PRACK or local codec policies.

For example:

1. The originating MSC issues an INVITE with an SDP offer of EVRCO (96).
2. The Oracle Enterprise Session Border Controller responds with EVRCO (96) in the SDP answer.
3. The far end puts the established call on hold, and subsequently resumes the call with a RE-INVITE.
4. The originating network policy adds EVCRO (97), incrementing the PT.
5. The originating MSC accepts the offered payload, EVRCO (97),but still answers with the originally negotiated PT, EVCRO (96).
6. Since the Oracle Enterprise Session Border Controller does not support asymmetric payloads, it accepts EVRCO (96) for both RTP flows, and sets up digital signal processors (DSPs) with these parameters.
7. However since the originating MSC received the EVCRO (97) and responded with EVCRO (96), it expects to receive RTP with PT=96, and send RTP with PT=97.
8. Consequently, the origination-side RTP is broken because the Oracle Enterprise Session Border Controller drops the RTP it receives with an unexpected PT=97.

To address this problem, this version supports asymmetric payload types such that RTP offered with one payload type and answered with another payload type is acceptable to the Oracle Enterprise Session Border Controller when providing transcoding.

Configure Transcoding for Asymmetric Dynamic Payload Types

Transcoding support for asymmetric dynamic payload types enables the Oracle Enterprise Session Border Controller to perform transcoding when the Real-time Transport Protocol (RTP) is offered with one payload type and is answered with another payload type. Enable transcoding for asymmetric dynamic payload types from the command line.

Before You Begin

- Confirm that you are in Superuser mode.

Procedure

1. Access the **media-manager-config** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

2. Type select to begin editing.

```
ACMEPACKET(media-manager-config)# select

ACMEPACKET(media-manager-config)#
```

3. Use the options +audio-allow-asymmetric-pt command to enable support for asymmetric payload types.

```
ACMEPACKET#(media-manager) options +audio-allow-asymmetric-pt
ACMEPACKET#(media-manager)
```

4. Type **done** to save your configuration.

Maintenance and Troubleshooting

show mbcd errors

The show mbcd errors command displays statistics related to MBCD task errors. The following fields are explained:

- XCode Internal Errors—Number of uncategorized errors due to Transcoding session error.
- XCode Alloc Errors—Number of times that buffer allocation failed for transcoding tasks.
- XCode Update Errors—Number of errors encountered when attempting to update an entry in the Transcoding table upon receipt of the first packet for a media flow.
- XCode Delete Errors—Number of errors encountered when attempting to delete an entry in the Transcoding table.
- XCode Over Cap Errors—Number of Transcoding sessions denied once session capacity is reached.
- XCode Over License Cap—Number of Transcoding sessions denied once license capacity is reached.

```
ACMEPACKET# show mbcd errors
13:22:50-126
MBC Errors/Events              ---- Lifetime ----
                        Recent      Total  PerMax
Client Errors                0          0       0
Client IPC Errors            0          0       0
Open Streams Failed          0          0       0
Drop Streams Failed          0          0       0
Exp Flow Events              0          0       0
Exp Flow Not Found           0          0       0
Transaction Timeouts         0          0       0
Server Errors                0          0       0
Server IPC Errors            0          0       0
Flow Add Failed            180        180     180
Flow Delete Failed           0          0       0
Flow Update Failed           0          0       0
Flow Latch Failed            0          0       0
Pending Flow Expired         0          0       0
ARP Wait Errors              0          0       0
Exp CAM Not Found            0          0       0
Drop Unknown Exp Flow        0          0       0
Drop/Exp Flow Missing        0          0       0
Exp Notify Failed            0          0       0
Unacknowledged Notify        0          0       0
Invalid Realm                0          0       0
No Ports Available           0          0       0
Insufficient Bandwidth       0          0       0
Stale Ports Reclaimed        0          0       0
Stale Flows Replaced         0          0       0
Telephone Events Gen         0          0       0
Pipe Alloc Errors            0          0       0
Pipe Write Errors            0          0       0
Not Found In Flows           0          0       0
XCode Internal Errors        0          0       0
XCode Alloc Errors           0          0       0
XCode Update Errors          0          0       0
XCode Delete Errors          0          0       0
XCode Over Cap Errors      180        180     180
XCode Over License Cap       0          0       0
SRTP Capacity Exceeded       0          0       0
```

show xcode api-stats

The show xcode api-stats command shows the client and server side message counts for the XClient and XServer software components. The main messages are allocate, update, and free of the transcoding resource. The command uses a 100 second window to show recent counts within the sliding window as well as the total and per max

(maximum in a sliding window interval). This command is useful for comparing the client and server side counts and seeing where errors may have occurred with the transcoding resources.

```
ACMEPACKET#show xcode api-stats
                      --------- Client --------   --------- Server --------
Message/Event         Recent      Total  PerMax   Recent      Total  PerMax
                      ------  ---------  ------   ------  ---------  ------
Allocs                     0       5197    4897        0       6355    6055
Updates                    0       1776    1676        0        888     788
Frees                      0       6355    6015        0       6355    6015
Error-Allocs               0          0       0        0         45      45
Error-Updates              0          0       0        0        888     888
Error-Frees                0          0       0        0          0       0
Total                      0      13328   12588        0      14531   13791
```

show xcode dbginfo

The debug information command shows the packet API statistics for the host to DSP path. There is one session/ connection opened with each DSP. The command displays the total packet counts as well as the round trip time statistics for the packets. The recent field shows the count since the last time the command was executed

```
ACMEPACKET#show xcode dbginfo
Startup Time    : 2006-09-08 01:11:50.522
Last Clear Time : 2006-09-08 01:11:50.522
Last Read Time  : 2006-09-08 17:14:52.351
Current Time    : 2006-09-08 17:14:52.351
Up Time         : 0 Days, 16 Hours 3 Minutes 2 Seconds
                          -- Life Time --    -- Recent --
PktApiStats:
    OpenConnectionCnt    =       2
    OpenSessionCnt       =       2
    TotalPktSentCnt      =   21051               21051
    TotalPktRecvCnt      =   21041               21041
    TotalPktRecvEventCnt =       0                   0
    TotalPktRecvDataCnt  =       0                   0
    TotalPktRejectCnt    =       5                   5
    TotalPktTimeoutCnt   =       0                   0
    TotalPktInvalidCnt   =       0                   0
    TotalPktDropCnt      =       0                   0
    TotalPktDropEventCnt =       0                   0
    TotalPktDropDataCnt  =       0                   0
    TotalPktLateRspCnt   =       0                   0
    LowestRoundTripMs             =       1
    HighestRoundTripMs            =    2010
    LowestExtractTimeMs           =       1
    HighestExtractTimeMs          =   13320
    HighestTransportRxTimeMs      =       0
    ulHighestTransportNoRxTimeMs  =       0
```

show sipd codecs

The **show sipd codecs <realm ID>** command displays media-processing statistics per SIP traffic. This command displays statistics per realm and requires a realm argument.

Session Based Statistics

The first 3 statistics listed by the **show sipd codecs** are session based. These statistics are titled Transcoded, Transrated, and Transparent.

- transcoded—counts of sessions that use the Transcoding NIU's TCUs to transcode between two or more codes.
- transrated—counts of sessions that use the Transcoding NIU's TCUs to change the packetization interval among dialogs in the session.

- transparent—counts of sessions that require no TCU hardware intervention (all end-to-end media uses the same codec)

A value of "none" which is not counted in the statistics is set when there is no media at all or media is not yet negotiated. Sessions within the same realm are counted only once.

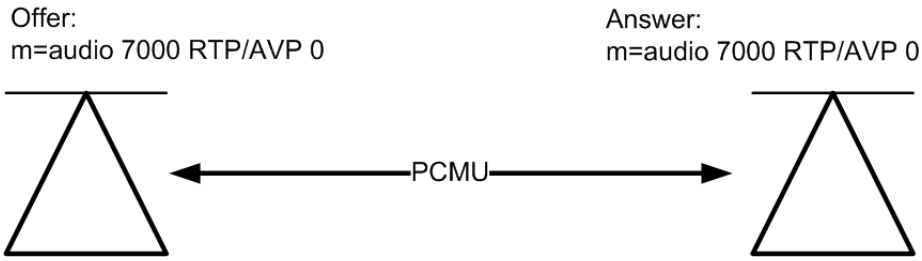
These are meter type counters, and thus have an "active" count as well as total lifetime values. The media-processing state of the session only can increase in precedence (highest=transcoded, transrated, transparent, lowest=none). Thus, if a session begins as transcoded, and then is re-negotiated to transparent later by a re-INVITE, it is still considered transcoded. However, if a session begins as transparent, it can go to transcoded by a re-INVITE. In such a case, the total counts for both transparent and transcoded would be incremented. If there are several media lines, the highest precedence is used for the session.

Flow Based Statistics

The remaining lines of the **show sipd codecs** command track the number of codecs in established sessions. The 'Other' type refers to unknown codecs. Only the Recent-Total and other lifetime columns are populated; the Active and Recent High are not applicable. Theses counts represent each SDP m= line emanating in the queried realm. Refer to the following examples:

Single audio stream example

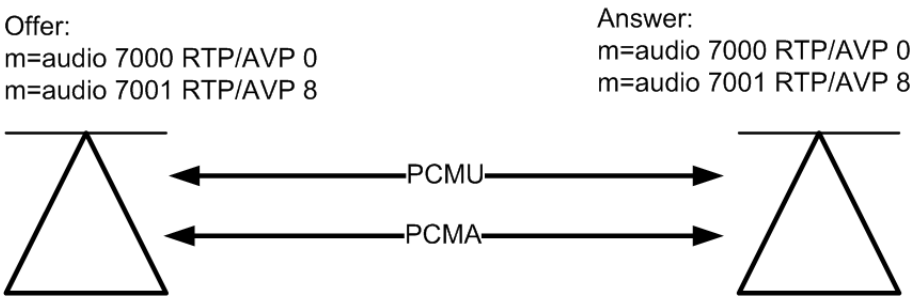
The following diagram shows an intra-realm session with one audio stream using the PCMU codec. Once the session is established, the PCMU count in the show sidp codecs output is 2.



If the session originator and terminator in the previous diagram exist in two different realms, you must execute the show sidp codecs command twice, once for each realm. A single PCMU count will be reflected in each respective query because only one m= line emanates from each realm.

Multiple audio stream example

The following diagram shows an intra-realm session with two audio streams. Each stream uses a different codec. Once the session is established, the PCMU count in the show sidp codecs output is 2, and the PCMA count is 2.

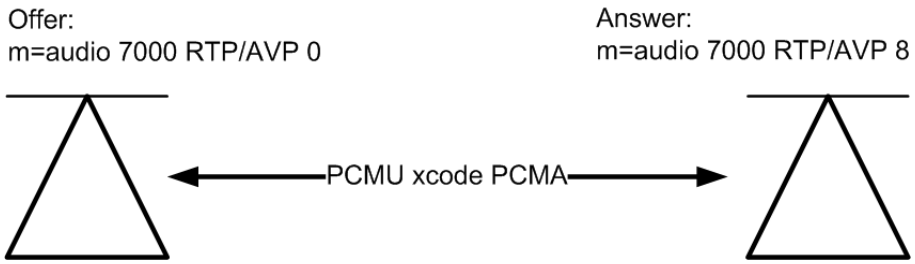


If the session originator and terminator in the previous diagram exist in two different realms, you must execute the show sidp codecs command twice, once for each realm. A single PCMU count and a single PCMA count will be reflected in each respective query because two m= lines emanate from each realm.

Transcoded audio stream example

The following diagram shows an intra-realm transcoding scenario where the originator and terminator are using different audio codecs. The Oracle Enterprise Session Border Controller transcodes the media, which is invisible to

the endpoints. Once the session is established, the PCMU count in the show sidp codecs output is 1, and the PCMA count is 1.



If the session originator and terminator in the previous diagram exist in two different realms, you must execute the show sidp codecs command twice, once for each realm. A single PCMU count appears in one query and a single PCMA count appears in the other query because only one m= line emanate from each realm.

An example show sidp codecs <realm ID> command follows:

```
ACMEPACKET# show sipd codecs realm01
18:30:41-595 Realm realm01
Codec Statistics
                         ---- Recent ---- -------- Lifetime --------
                    Active    High   Total      Total  PerMax    High
Transcoded               0       0       0          0       0       0
Transrated               0       0       0          0       0       0
Transparent              0       0       0          0       0       0
PCMU Count               -       -       0          0       0
PCMA Count               -       -       0          0       0
G722 Count               -       -       0          0       0
G723 Count               -       -       0          0       0
G726-16 Count            -       -       0          0       0
G726-24 Count            -       -       0          0       0
G726-32 Count            -       -       0          0       0
G726-40 Count            -       -       0          0       0
G728 Count               -       -       0          0       0
G729 Count               -       -       0          0       0
GSM Count                -       -       0          0       0
iLBC Count               -       -       0          0       0
H261 Count               -       -       0          0       0
H263 Count               -       -       0          0       0
T38 Count                -       -       0          0       0
AMR Count                -       -       0          0       0
AMR-WB Count             -       -       0          0       0
EVRC Count               -       -       0          0       0
Other Count              -       -       0          0       0
```

show xcode load

The show xcode load command shows the current transcoding module (TCM) load both in number of sessions and percent loading. The load percentage depends on the precise mix of codecs, ptimes, and features enabled on the active sessions. The maximum lifetime load is also displayed. Uninstalled TCMs are marked with dashes.

```
ACMEPACKET#show xcode load
17:19:11
Total Sessions:              0
Licensed AMR Sessions:       0
Licensed AMR-WB Sessions:    0
Licensed EVRC Sessions:      0
Licensed EVRCB Sessions:     0
                    ----- Load -----
  TCU  TCM  #Sess  Current  Maximum
  ===  ===  =====  =======  =======
   0    00     -       -        -
```

```
   0    01     -       -        -
   0    02     -       -        -
   0    03     0     0.00%   99.81%
   0    04     0     0.00%   99.81%
   0    05     -       -        -
   0    06     -       -        -
   0    07     -       -        -
```

The TCU column is populated with a 0 for a TCM in the middle slot and a 1 for a TCM in the top slot.

show xcode session-all

The **show xcode session-all** command displays all of the currently active sessions by their unique session id.

```
ACMEPACKET#show xcode session-all
15:22:51
Requesting xclient sessions table
        Total Active Sessions: 200
        Displaying sessions 1 to 100:
        Session Id: 0x10007
        Session Id: 0x10008
        Session Id: 0x10009
        Session Id: 0x1000a
        Session Id: 0x1000b
```

Note: When there are more than 100 active sessions , the command now displays only active sessions 1 to 100 as opposed to all the active session:

show xcode session-byid

The session-byid command gives more detailed information about the session. The session-byid command displays the configuration of each channel as well as a number of packet statistics for each channel. This same information can be looked up by IP address and port by using the session-byip command. If only the configuration portion is required, use the session-config command with the session id as the argument. This command is entered as:

```
show xcode     session-byid <session_id>
```

For example:

```
ACMEPACKET#show xcode session-byid 0xf006e
################## SESSION  0xf006e ##################
Channel 0:
  DSP device          = 14
  Source MAC          = 00:08:25:a0:9a:f3
  Destination MAC     = 00:0e:0c:b7:32:e2
  VLAN ID             = 0
  Egress Interface    = 0
  Src IP:Port         = 172.16.0.235:24448
  Dst IP:Port         = 172.16.0.87:16000
  Src RTCP IP:Port    = 172.16.0.235:24449
  Dst RTCP IP:Port    = 172.16.0.87:16001
  Codec               = G711_ULAW_PCM
  Payload Type        = 0
  Pkt Interval        = 20 msec
  2833 Payload Type   = DISABLED
  Xtone Mode          = XTONE_XTHRU
  Status              = DISABLED
DSP Counters:
  RxInPktCnt                  474
  RxInByteCnt                 75840
  RxOutPktCnt                 749
  RxInSidPktCnt               0
  RxNoPktCnt                  275
  RxBadPktTypeCnt             0
  RxBadRtpPayloadTypeCnt      0
  RxBadPktHdrFormatCnt        0
```

```
  RxBadPktLengthCnt           0
  RxMisorderedPktCnt          0
  RxBadPktChecksumCnt         0
  RxUnderrunSlipCnt           0
  RxOverrunSlipCnt            0
  RxLastVocoderType           0
  RxVocoderChangeCnt          0
  RxMaxDetectedPdv            168
  RxDecdrRate                 15
  RxJitter:
    CurrentDelay              160
    EstimatedDelay            0
    ClkDriftingDelta          0
    ClkDriftingCorrectionCnt  0
    InitializationCnt         1
  RxCircularBufferWriteErrCnt 0
  RxApiEventCnt               0
  TxCurrentVocoderType        0
  TxInPktCnt                  749
  TxOutPktCnt                 750
  TxOutByteCnt                120000
  TxInBadPktPayloadCnt        0
  TxTimestampGapCnt           0
  TxTdmWriteErrCnt            0
  RxToneDetectedCnt           0
  RxToneRelayEventPktCnt      0
  RxToneRelayUnsupportedCnt   0
  TxToneRelayEventPktCnt      0
  TxApiEventCnt               0
  TxNoRtpEntryPktDropCnt      0
  ConnectionWaitAckFlag       1
  RxMipsProtectionDropCnt     0
  TxMipsProtectionDropCnt     0
Channel 1:
  DSP device          = 14
  Source MAC          = 00:08:25:a0:9a:f4
  Destination MAC     = 00:1b:21:7a:29:b1
  VLAN ID             = 0
  Egress Interface    = 2
  Src IP:Port         = 192.168.0.235:24448
  Dst IP:Port         = 192.168.0.87:32000
  Src RTCP IP:Port    = 192.168.0.235:24449
  Dst RTCP IP:Port    = 192.168.0.87:32001
  Codec               = G729_A
  Payload Type        = 18
  Pkt Interval        = 20 msec
  2833 Payload Type   = DISABLED
  Xtone Mode          = XTONE_XTHRU
  Status              = DISABLED
DSP Counters:
  RxInPktCnt                  748
  RxInByteCnt                 14960
  RxOutPktCnt                 751
  RxInSidPktCnt               0
  RxNoPktCnt                  3
  RxBadPktTypeCnt             0
  RxBadRtpPayloadTypeCnt      0
  RxBadPktHdrFormatCnt        0
  RxBadPktLengthCnt           0
  RxMisorderedPktCnt          0
  RxBadPktChecksumCnt         0
  RxUnderrunSlipCnt           0
  RxOverrunSlipCnt            0
  RxLastVocoderType           6
```

```
  RxVocoderChangeCnt          0
  RxMaxDetectedPdv            171
  RxDecdrRate                 15
  RxJitter:
    CurrentDelay              160
    EstimatedDelay            0
    ClkDriftingDelta          0
    ClkDriftingCorrectionCnt  0
    InitializationCnt         1
  RxCircularBufferWriteErrCnt 0
  RxApiEventCnt               0
  TxCurrentVocoderType        6
  TxInPktCnt                  748
  TxOutPktCnt                 748
  TxOutByteCnt                14960
  TxInBadPktPayloadCnt        0
  TxTimestampGapCnt           0
  TxTdmWriteErrCnt            0
  RxToneDetectedCnt           0
  RxToneRelayEventPktCnt      0
  RxToneRelayUnsupportedCnt   0
  TxToneRelayEventPktCnt      0
  TxApiEventCnt               0
  TxNoRtpEntryPktDropCnt      0
  ConnectionWaitAckFlag       0
  RxMipsProtectionDropCnt     0
  TxMipsProtectionDropCnt     0
```

show xcode session-byattr

The show xcode session-byattr command lists all sessions matching the specified attribute name. The only supported attribute is "fax", which will display session information only for FAX-transcoded sessions; all other attributes will return an error. For example:

```
ACMEPACKET#show xcode session-byattr fax
17:52:17
1.   [Chan A_SRC] Ip Address: 192.168.16.1 Port: 5220
     [Chan B_SRC] Ip Address: 172.16.0.40 Port: 10230
     Session Id:0x002021d0"
2.   [Chan A_SRC] Ip Address: 192.168.16.1 Port: 3010
     [Chan B_SRC] Ip Address: 172.16.0.40 Port: 10226
     Session Id:0x00301042"
Oct 26 20:53:22.968 Total Matches:2 Total Active Sessions:2
-------------
```

show xcode session-byipp

The show xcode session-byipp command requires an IP address and port. It lists detailed information about the sessions identified by the specified IP address and port number. nformation will be provided for all transcoded call legs matching the IP address, including in both the ingress and egress directions. The show xcode session-byipp command is entered as:

```
show xcode session-byipp <ip_addr> <port_num>
```

This command displays the same information as the session-byid command. If a wildcard * is provided for the port number, the command will display sessions with the matching IP address only, regardless of port number.

show xcode xlist

The show xcode xlist command displays the TCU (0 = middle, 1 = top), TCM number, number of DSPs on each module, the number of active sessions, and the load percentage. It also displays the state such as Active or Boot Failure. Uninstalled TCMs are indicated by a dash.

```
ACMEPACKET#show xcode xlist
18:22:32
```

```
 TCU  TCM  DSPs  #Sess  Load   State
 ===  ===  ====  =====  ====  =========
  0    00    2      -     -      -
  0    01    2      -     -      -
  0    02    2      -     -      -
  0    03    2      1     0%   2 Active
  0    04    2      1     0%   2 Active
  0    05    2      -     -      -
  0    06    2      -     -      -
  0    07    2      -     -      -
  [...]
  1    00    2      1     0%   2 Active
  1    01    2      0     0%   2 Active
  1    02    2      0     0%   2 Active
  1    03    2      0     0%   2 Active
  1    04    2      0     0%   2 Active
  1    05    2      0     0%   2 Active
  1    06    2      0     0%   2 Active
  1    07    2      0     0%   2 Active
  1    08    2      0     0%   2 Active
```

show xcode load for software xcode

The **show xcode load** command displays the total number of sessions, the number of licensed G729 sessions that are currently active, and transcoding processing load. Each row that corresponds to a transcoding module (TCM) is a single software transcoding thread. Percentage values per thread are calculated based on the total number of available sessions.

```
# show xcode load
12:15:54
Total Sessions:              0
Licensed G729 Sessions:      0
                      ----- Load -----
         ID  #Sess  Current  Maximum
         ==  =====  =======  =======
TCM    :  0      0    0.00%    0.00%
TCM    :  1      0    0.00%    0.24%
```

show xcode session-all

The **show xcode session-all** command displays all of the currently active sessions by their unique session id.

```
ACMEPACKET#show xcode session-all
15:22:51
Requesting xclient sessions table
        Total Active Sessions: 200
        Displaying sessions 1 to 100:
        Session Id: 0x10007
        Session Id: 0x10008
        Session Id: 0x10009
        Session Id: 0x1000a
        Session Id: 0x1000b
```

Note: When there are more than 100 active sessions , the command now displays only active sessions 1 to 100 as opposed to all the active session:

show xcode session-byid

The **show xcode session-byid** command displays detailed information about the supplied software based transcoding session. The session-byid command displays the configuration of each channel as well as a number of packet statistics for each channel. This same information can be looked up by IP address and port by using the **show xcode session-byip** command. If only the configuration portion is required, use the session-config command with the session id as the argument.

This command is entered as:

```
# show xcode session-byid 0x10001
17:53:01
################## SESSION 0x10001 ##################
Channel 0:
  HSP device          = 0
  Egress Interface    = 0
  Src IP:Port         = 172.16.0.241:10000
  Dst IP:Port         = 172.16.0.15:6000
  Codec               = G711_ALAW_PCM
  Payload Type        = 8
  Pkt Interval        = 20 msec
  Status              = ENABLED

HSP Counters:
  RxPackets               996
  RxLatePackets           0

  RxMisorderedPackets     0
  RxBadPackets            0
  RxEarlyPackets          0
  RxDroppedPackets        0
  RxFailures              0

  TxPackets               993
  TxFailures              0

Channel 1:
  HSP device          = 0
  Egress Interface    = 1
  Src IP:Port         = 192.168.0.241:10000
  Dst IP:Port         = 192.168.0.15:6000
  Codec               = G711_ULAW_PCM
  Payload Type        = 0
  Pkt Interval        = 20 msec
  Status              = ENABLED

HSP Counters:
  RxPackets               996
  RxLatePackets           0

  RxMisorderedPackets     0
  RxBadPackets            0
  RxEarlyPackets          0
  RxDroppedPackets        0
  RxFailures              0

  TxPackets               993
  TxFailures              0
```

show xcode session-byipp

The show xcode session-byipp command requires an IP address and port. It lists detailed information about the sessions identified by the specified IP address and port number. nformation will be provided for all transcoded call legs matching the IP address, including in both the ingress and egress directions. The show xcode session-byipp command is entered as:

```
show xcode session-byipp <ip_addr> <port_num>
```

This command displays the same information as the session-byid command. If a wildcard * is provided for the port number, the command will display sessions with the matching IP address only, regardless of port number.

Logs

A log file named log.xserv can be used for debugging the transcoding feature. This log records the API between the host software and the DSPs and any errors that are encountered.

Alarms

The transcoding feature employs several hardware and software alarms to alert the user when the system is not functioning properly or overload conditions are reached.

Name/ID	Severity/ Health Degredation	Cause(s)	Traps Generated
No DSPs Present with Transcoding Feature Card (DSP_NONE_PRESENT)	Minor/0	A transcoding feature card is installed but no DSP modules are discovered.	apSysMgmtHardwareErrorTrap
DSP Boot Failure (DSP_BOOT_FAILURE)	Critical/0	A DSP device fails to boot properly at system initialization. This alarm is not health affecting for a single DSP boot failure. DSPs that fail to boot will remain uninitialized and will be avoided for transcoding.	apSysMgmtHardwareErrorTrap
DSP Communications Timeout (DSP_COMMS_TIMEOUT)	Critical/100	A DSP fails to respond after 2 seconds with 3 retry messages. This alarm is critical and is health affecting.	apSysMgmtHardwareErrorTrap
DSP Alerts (DSP_CORE_HALT)	Critical/100	A problem with the health of the DSP such as a halted DSP core. The software will attempt to reset the DSP and gather diagnostic information about the crash. This information will be saved in the /code directory to be retrieved by the user.	apSysMgmtHardwareErrorTrap
DSP Temperature(DSP_TEMPERATURE_HIGH)	Clear 85°C Warning 86°C / 5 Minor 90°C / 25 Major 95°C/ 50 Critical 100°C/ 100	A DSP device exceeds the temperature threshold. If the temperature exceeds 90°C, a minor alarm will be set. If it exceeds 95°C, a major alarm will be set. If it exceeds 100°C, a critical alarm will be set. The alarm is cleared if the temperature falls below 85°C. The alarm is health affecting.	apSysMgmtHardwareErrorTrap

Name/ID	Severity/ Health Degredation	Cause(s)	Traps Generated
Transcoding Capacity Threshold Alarm (XCODE_UTIL_OVER_THRESHOLD) / 131329	Clear 80% Warning 95%	A warning alarm will be raised when the transcoding capacity exceeds a high threshold of 95%. The alarm will be cleared after the capacity falls below a low threshold of 80%. This alarm warns the user that transcoding resources are nearly depleted. This alarm is not health affecting.	apSysMgmtGroupTrap
Licensed AMR Transcoding Capacity Threshold Alarm/ 131330	Clear 80% Warning 95%	A warning alarm is triggered if the AMR transcoding capacity exceeds a high threshold of 95% of licensed session in use. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	apSysMgmtGroupTrap
Licensed AMR-WB Transcoding Capacity Threshold Alarm/ 131331	Clear 80% Warning 95%	A warning alarm is triggered if the AMR-WB transcoding capacity exceeds a high threshold of 95% of licensed session in use. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	apSysMgmtGroupTrap
Licensed EVRC Transcoding Capacity Threshold Alarm/ 131332	Clear 80% Warning 95%	A warning alarm is triggered if the EVRC transcoding capacity exceeds a high threshold of 95% of licensed session in use. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	apSysMgmtGroupTrap
Licensed EVRCB Transcoding Capacity Threshold Alarm/ 131333	Clear 80% Warning 95%	A warning alarm is triggered if the EVRCB transcoding capacity exceeds a high threshold of 95% of licensed session in use. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	apSysMgmtGroupTrap

Transcoding Capacity Traps

The Oracle Enterprise Session Border Controller sends the apSysMgmtGroupTrap as transcoding capacity nears its limit. This trap is sent and cleared for three conditions:

- Total DSP usage exceeds 95%
- Total AMR sessions exceed 95% of licensed capacity
- Total AMR-WB sessions exceed 95% of licensed capacity
- Total EVRC sessions exceed 95% of licensed capacity
- Total EVRCB sessions exceed 95% of licensed capacity

The apSysMgmtGroupTrap contains the condition observed (apSysMgmtTrapType) and the corresponding value reached (apSysMgmtTrapValue).

```
apSysMgmtGroupTrap          NOTIFICATION-TYPE
    OBJECTS         { apSysMgmtTrapType, apSysMgmtTrapValue }
    STATUS            current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          exceeds a certain threshold. "
    ::= { apSystemManagementNotifications 1 }
```

When the resource usage retreats below a defined threshold, the Oracle Enterprise Session Border Controller sends an apSysMgmtGroupClearTrap.

```
apSysMgmtGroupClearTrap          NOTIFICATION-TYPE
    OBJECTS         { apSysMgmtTrapType }
    STATUS            current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          returns to within a certain threshold.  This signifies that
          an alarm caused by that monitoring object has been cleared. "
    ::= { apSystemManagementNotifications 2 }
```

The following table summarizes trigger and clear conditions for transcoding capacity alerts as sent in the the apSysMgmtGroupTrap:

Monitored Transcoding Resource	SNMP Object & OID in apSysMgmtTrapType	Trap Sent	Clear Trap Sent
Total DSP Usage	apSysXCodeCapacity 1.3.6.1.4.1.9148.3.2.1.1.34	95%	80%
AMR License Capacity Usage	apSysXCodeAMRCapacity 1.3.6.1.4.1.9148.3.2.1.1.35	95%	80%
AMR-WB License Capacity Usage	apSysXCodeAMRWBCapacity 1.3.6.1.4.1.9148.3.2.1.1.36	95%	80%
EVRC License Capacity Usage	apSysXCodeEVRCCapacity 1.3.6.1.4.1.9148.3.2.1.1.39	95%	80%
EVRCB License Capacity Usage	apSysXCodeEVRCBCapacity 1.3.6.1.4.1.9148.3.2.1.1.40	95%	80%

The following SNMP Objects are inserted into the apSysMgmtTrapType when sending and clearing a transcoding capacity trap. You mayt query them individually with an SNMP GET.

- apSysXCodeCapacity (1.3.6.1.4.1.9148.3.2.1.1.34)
- apSysXCodeAMRCapacity (1.3.6.1.4.1.9148.3.2.1.1.35)
- apSysXCodeAMRWBCapacity (1.3.6.1.4.1.9148.3.2.1.1.36)
- apSysXCodeEVRCCapacity (1.3.6.1.4.1.9148.3.2.1.1.39)
- apSysXCodeEVRCBCapacity(1.3.6.1.4.1.9148.3.2.1.1.40)

SNMP

Acme Packet Codec and Transcoding MIB (ap-codec.mib)

The following table describes the SNMP GET query names for the Oracle Codec and Transcoding MIB (ap-codec.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apCodecMIBObjects (1.3.6.1.4.1.9148.3.7.1)		
Object Identifier Name: apCodecRealmStatsTable (1.3.6.1.4.1.9148.3.7.1.1)		
Object Identifier Name: apCodecRealmStatsEntry (1.3.6.1.4.1.9148.3.7.1.1.1)		
apCodecRealmCountOther	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.1	Count of the SDP media streams received in the realm which negotiated to a codec not defined in this table.
apCodecRealmCountPCMU	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.2	Count of SDP media streams received in the realm which negotiated to the PCMU codec.
apCodecRealmCountPCMA	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.3	Count of SDP media streams reveived in the realm which negotiated to the PCMA codec.
apCodecRealmCountG722	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.4	Count of SDP media streams reveived in the realm which negotiated to the G722 codec.
apCodecRealmCountG723	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.5	Count of SDP media streams reveived in the realm which negotiated to the G723 codec.
apCodecRealmCountG726-16	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.6	Count of SDP media streams reveived in the realm which negotiated to the G726-16 codec.
apCodecRealmCountG726-24	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.7	Count of SDP media streams reveived in the realm which negotiated to the G726-24 codec.
apCodecRealmCountG726-32	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.8	Count of SDP media streams reveived in the realm which negotiated to the G726-32 codec.
apCodecRealmCountG726-40	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.9	Count of SDP media streams reveived in the realm which negotiated to the G726-40 codec.
apCodecRealmCountG728	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.10	Count of SDP media streams reveived in the realm which negotiated to the G728 codec.

SNMP GET Query Name	Object Identifier Name: Number	Description
apCodecRealmCountG729	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.11	Count of SDP media streams reveived in the realm which negotiated to the G729 codec.
apCodecRealmCountGSM	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.12	Count of SDP media streams reveived in the realm which negotiated to the GSM codec.
apCodecRealmCountILBC	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.13	Count of SDP media streams reveived in the realm which negotiated to the iLBC codec.
apCodecRealmCountAMR	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.14	Count of SDP media streams reveived in the realm which negotiated to the AMR codec.
apCodecRealmCountEVRC	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.15	Count of SDP media streams reveived in the realm which negotiated to the EVRC codec.
apCodecRealmCountH261	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.16	Count of SDP media streams reveived in the realm which negotiated to the H261 codec.
apCodecRealmCountH263	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.17	Count of SDP media streams reveived in the realm which negotiated to the H.263 codec.
apCodecRealmCountT38	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.18	Count of SDP media streams reveived in the realm which negotiated to the T.38 codec.
apCodecRealmCountAMRWB	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.19	Count of SDP media streams reveived in the realm which negotiated to the AMR-WB codec.
Object Identifier Name: apTranscodingMIBObjects (1.3.6.1.4.1.9148.3.7.2)		
Object Identifier Name: apCodecTranscodingRealmStatsTable (1.3.6.1.4.1.9148.3.7.2.1)		
Object Identifier Name: apTranscodingRealmStatsEntry (1.3.6.1.4.1.9148.3.7.2.1.1)		
apCodecRealmSessionsTransparent	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.1	Number of sessions in the realm that did not use any DSP resources for transcoding or transrating.
apCodecRealmSessionsTransrated	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.2	Number of sessions in the realm that had a common codec but used DSP resources to modify packetization rate.
apCodecRealmSessionsTranscoded	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.3	Number of sessions in the realm that had used DSP resources to transcode between codecs.
Object Identifier Name: apSysMgmtMIBSessionObjects (1.3.6.1.4.1.9148.3.2.1.2)		
Object Identifier Name: apSigRealmStatsTable (1.3.6.1.4.1.9148.3.2.1.2.4)		
Object Identifier Name: apSigRealmStatsEntry (1.3.6.1.4.1.9148.3.2.1.2.4.1)		

SNMP GET Query Name	Object Identifier Name: Number	Description
apSigRealmStatsRealmName	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.2	Nmae of the realm the following for which the following statistics are being calculated.
apSigRealmStatsCurrentActiveSessionsInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.3	Number of current active inbouind sessions.
apSigRealmStatsCurrentSessionRateInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.4	Current inbound session rate in CPS.
apSigRealmStatsCurrentActiveSessionsOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.5	Number of current active outbound sessions.
apSigRealmStatsCurrentSessionRateOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.6	Current outbound session rate in CPS.
apSigRealmStatsTotalSessionsInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.7	Total number of inbound sessions.
apSigRealmStatsTotalSessionsNotAdmittedInbouind	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.8	Total number of inbound sessions rejected due to insufficient bandwidth.
apSigRealmStatsPeriodHighInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.9	Highest number of concurrent inbound sessions during the period.
apSigRealmStatsAverageRateInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.10	Average rate of inbound sessions during the period in CPS.
apSigRealmStatsTotalSessionsOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.11	Total number of outbound sessions.
apSigRealmStatsTotalSessionsNotAdmittedOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.12	Total number of outbound sessions rejected due to insufficient bandwidth.
apSigRealmStatsPeriodHighOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.13	Highest number of concurrent outbound sessions during the period.
apSigRealmStatsAverageRateOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.14	Average rate of outbound sessions during the period in CPS.
apSigRealmStatsMaxBurstRate	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.15	Maximum burst rate of traffic measured during the period (combined inbound and outbound).

SNMP GET Query Name	Object Identifier Name: Number	Description
apSigRealmStatsPeriodSeizures	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.16	Total number of seizures during the period.
apSigRealmStatsPeriodAnswers	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.17	Total number of answered sessions during the period.
apSigRealmStatsPeriodASR	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.18	Answer-to-seizure ratio, expressed as a percentage. For example, a value of 90 represents 90% or .90.
apSigRealmStatsAverageLatency	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.19	Average observed one-way signaling latency during the period in milliseconds.
apSigRealmStatsMaxLatency	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.20	Maximum observed one-way signaling latency during the period in milliseconds.
apSigRealmStatsMinutesLeft	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.21	Number of montly-minutes left in the pool per calendar month for a given realm.
apSigRealmStatsMinutesReject	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.22	Peg counts of number of rejected calls due to monthly-minutes constraints exceeded.
apSigRealmStatsShortSessions	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.23	Lifetime number of sessions whose duration was less than the configured short session duration.

Acme Packet System Management MIB (ap-smgmt.mib)

The following VARBINDs are used in Transcoding related traps. They may not be polled and retrieved using an SNMP GET.

SNMP Object Name	Object Identifier Name: Number	Description
Object Identifier Name: apSysMgmtMIBObjects (1.3.6.1.4.1.9148.3.2.1)		
Object Identifier Name: apSysMgmtGeneralObjects (1.3.6.1.4.1.9148.3.2.1.1)		
apSysXCodeCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.34	Percentage of transcoding utilization.
apSysXCodeAMRCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.35	Percentage of licensed AMR transcoding sessions.

SNMP Object Name	Object Identifier Name: Number	Description
apSysXCodeAMRWBCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.36	Percentage of licensed AMR-WB transcoding sessions.
apSysXCodeEVRCCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.39	Percentage of licensedEVRC transcoding sessions.
apSysXCodeEVRCBCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.39	Percentage of licensedEVRCB transcoding sessions.

21

Communications Monitoring Probe

Palladion Mediation Engine

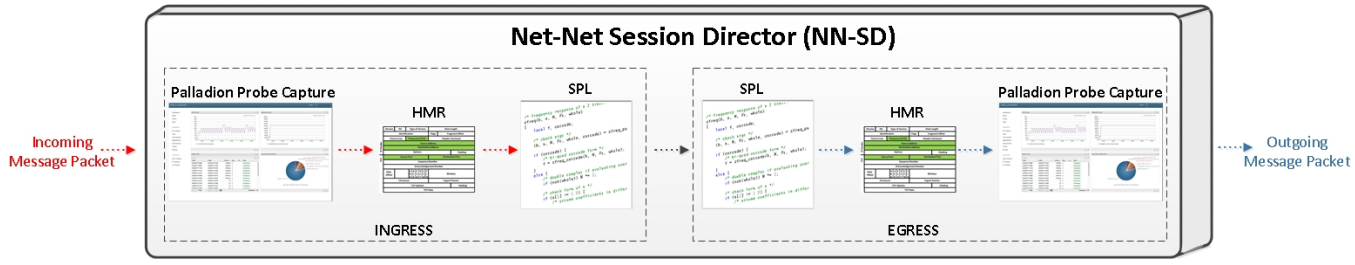
The Palladion Mediation Engine powers the Experience Manager for the Oracle Enterprise Session Border Controller.

The Experience Manager is a platform that collects SIP, DNS, ENUM, and protocol message traffic received from Palladion Probes. The mediation engine stores the traffic in an internal database, and analyzes aggregated data to provide comprehensive multi-level monitoring, troubleshooting, and interoperability information.

The Oracle Enterprise Session Border Controller supports an embedded, user-configurable Palladion Communications Monitoring Probe that can act as a probe or as an exporter. The Oracle Enterprise Session Border Controller can:

- Establish an authenticated, persistent, reliable TCP connection between itself and one or more Palladion Mediation Engines.
- Ensure message privacy by encrypting the TCP connection using TLS.
- Use the TCP connection to send a UTC-timestamped, unencrypted copy of a protocol message to a Palladion Engine.
- Accompany the copied message with related data to include the port/vlan on which the message was sent/ received, local and remote IP:port information, and the transport layer protocol.

The following illustration shows how the Palladion Communications Monitor Probe handles incoming and outgoing monitored data on the Net-Net ESD.



Note: For large TCP packets, the minimum required version of Oracle Communications Operations Monitor (OCOM) is 3.3.70.

Palladion Mediation Engine on Different Sub-nets

The Palladion Mediation Engine simplifies the operation of software-based Palladion probes by enabling the transmission of Internet Protocol Flow Information Export (IPFIX) data to one or more Palladion Mediation Engines.

When deployed on different sub-nets, the ACLI hierarchy is different because the network-interface parameter is removed from the comm-monitor configuration object and is transferred to the monitor-collector configuration object.

When migrating from the S-C[xz]6.3.9 release to the E-C[xz]6.4.0 release, probes anchored on media interfaces revert to the default network-interface value of wancom0:0.

The following illustration shows an S-C(xz)6.3.9 configuration.

```
comm-monitor
    state    enabled
    qos-enable    disabled
    sbc-grp-id    0
    tls-profile
    network-interface    M10:0
    monitor-collector
            address    172.16.29.102
            port    4739
```

The following illustration shows the upgraded E-C[xz]6.4.0 configuration.

```
comm-monitor
    state        enabled
    qos-enable        disabled
    sbc-grp-id        0
    tls-profile
    monitor-collector
        address    172.16.29.102
        port    4739
        network-interface    wancom0:0
```

Note: To restore prior service, update the network-interface parameter to its original value of M10:0.

Communications Monitor Configuration

Communications Monitor configuration consists of the following steps.

1. Configuration of one or more Oracle Enterprise Session Border Controller/Palladion exporter/collector pairs.

   Configuration of the -config object, which defines common operations across all interfaces, is not required. Default values can be retained to enable standard service.
2. Optional assignment of a TLS profile to an exporter/collector pair.

   Note: The Palladion Communications Monitor Probe communicates over the media interface for signaling and Quality of Service (QoS) statistics using IPFIX. QoS reporting is done via Call Detail Records (CDR) (accounting).

Communication Monitor

Use the following procedure to configure communication monitoring:

1. From superuser mode, use the following ACLI sequence to access comm-monitor configuration mode. From comm-monitor mode, you establish a connection between the Oracle Enterprise Session Border Controller, acting as a exporter of protocol message traffic and related data, and an Oracle® Communications Session Monitor Mediation Engine, acting as an information collector.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# system
   ACMEPACKET(system)# system-config
   ACMEPACKET(system-config)# comm-monitor
   ACMEPACKET(comm-monitor)#
   ```
2. Use the state parameter to enable or disable communication monitoring.

   Communication monitoring is disabled by default.

```
ACMEPACKET(comm-monitor)# state enabled
ACMEPACKET(comm-monitor)#
```

3. Use the sbc-group-id parameter to assign an integer value to the Oracle Enterprise Session Border Controller, in its role as an information exporter.

   Retain the default value (0) or assign another integer value.

```
ACMEPACKET(comm-monitor)# sbc-group-id 5
ACMEPACKET(comm-monitor)#
```

4. If the network interface specified in Step 8 is a media interface, you can optionally use TLS to encrypt the exporter/collector connection.

   To enable TLS encryption, use the tls-profile parameter to identify a TLS profile to be assigned to the network interface. The absence of an assigned TLS profile (the default state) results in unencrypted transmission.

   Refer to *TLS Profile Configuration* for configuration details.

```
ACMEPACKET(comm-monitor)# tls-profile commMonitor
ACMEPACKET(comm-monitor)#
```

5. Use the qos-enable parameter to enable or disable to export of RTP, SRTP, and QOS data flow information.

```
ACMEPACKET(comm-monitor)# qos-enable enabled
ACMEPACKET(comm-monitor)#
```

6. Use the monitor-collector parameter to move to monitor-collector configuration mode.

   While in this mode you identify a Communications Session Monitor Mediation Engine collector by IP address and port number.

```
ACMEPACKET(comm-monitor)# monitor-collector
ACMEPACKET(monitor-collector)#
```

7. Use the address and port parameters to specify the IP address and port number monitored by a Communications Session Monitor Mediation Engine for incoming IPFIX traffic.

   Enter an IPv4 address and a port number with the range 1025 through 65535, with a default value of 4739.

```
ACMEPACKET(monitor-collector)# address 172.30.101.239
ACMEPACKET(monitor-collector)# port 4729
ACMEPACKET(monitor-collector)#
```

8. Use the network-interface parameter to specify the network interface that supports the TCP connection between the Oracle Enterprise Session Border Controller to the Communications Session Monitor Mediation Engine.

   To specify the wancom0 management interface:

```
ACMEPACKET(comm-monitor)# network-interface wancom0:0
ACMEPACKET(comm-monitor)#
```

   To specify a media interface:

```
ACMEPACKET(comm-monitor)# network-interface m01
ACMEPACKET(comm-monitor)#
```

   Note: If configuring with a media interface, that interface must belong to a configured realm.

9. Use done and exit to return to comm-monitor configuration mode.
10. Use done, exit, and verify-config to complete configuration.
11. Repeat Steps 1 through 10 to configure additional as required.

TLS Profile Configuration

Use the following procedure to configure a tls-profile that identifies the cryptographic resources, specifically certificates and protocols, required for the establishment of a secure/encrypted connection between the Oracle Enterprise Session Border Controller and the Communications Session Monitor Mediation Engine.

1. From superuser mode, use the following command sequence to access tls-profile configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)#
```

2. Use the name parameter to provide a unique identifier for this tls-profile.

```
ACMEPACKET(tls-profile)# name commMonitor
ACMEPACKET(tls-profile)#
```

3. Use the required end-entity-certificate parameter to specify the name of the certificate-record configuration that identifies the credential (specifically, an X509.v3 certificate) offered by the Oracle Enterprise Session Border Controller in support of its asserted identity.

```
ACMEPACKET(tls-profile)# end-entity-certificate commMonitor509
ACMEPACKET(tls-profile)#
```

4. Use the required trusted-ca-certificates parameter to compile a list or one or more certificate-record configuration elements referencing trusted Certification Authority (CA) certificates used to authenticate the offered certificate. These referenced certificates are conveyed to the Communications Session Monitor Mediation Engine as part of the TLS exchange.

   Provide a comma separated list of existing CA certificate-record configuration elements.

```
ACMEPACKET(tls-profile)# trusted-ca-certificates verisignClass3-
a,verisignClass3-b,baltimore,thawtePremium,acme-CA
ACMEPACKET(tls-profile)#
```

5. Retain the default value, all, for the cipher-list parameter.
6. Use the verify-depth parameter to specify the maximum number of chained certificates that will be processed while authenticating end-entity certificate received from the Communications Session Monitor Mediation Engine.

   Provide an integer within the range 1 through 10 (the default).

   The Oracle Enterprise Session Border Controller supports the processing of certificate chains (consisting of an end-entity certificate and some number of CA certificates) when X.509v3 certificate-based authentication is used. The following process validates a received TLS certificate chain.

   - Check the validity dates (Not Before and Not After fields) of the end certificate. If either date is invalid, authentication fails; otherwise, continue chain validation
   - Check the maximum length of the certificate chain (specified by verify-depth). If the current chain exceeds this value, authentication fails; otherwise, continue chain validation.
   - Verify that the Issuer field of the current certificate is identical to the Subject field of the next certificate in the chain. If values are not identical, authentication fails; otherwise, continue chain validation.
   - Check the validity dates (Not Before and Not After fields) of the next certificate. If either date is invalid, authentication fails; otherwise, continue chain validation.
   - Check the X509v3 Extensions field to verify that the current certificate identifies a CA. If not so, authentication fails; otherwise, continue chain validation.
   - Extract the Public Key from the current CA certificate. Use it to decode the Signature field of the prior certificate in the chain. The decoded Signature field yields an MD5 hash value for the contents of that certificate (minus the Signature field).
   - Compute the same MD5 hash. If the results are not identical, authentication fails; otherwise, continue chain validation.
   - If the hashes are identical, determine if the CA identified by the current certificate is a trust anchor by referring to the trusted-ca-certificates attribute of the associated TLS-profile configuration object. If the CA is trusted, authentication succeeds. If not, return to Step 2.

```
ACMEPACKET(tls-profile)# verify-depth 8
ACMEPACKET(tls-profile)#
```

7. Use the mutual-authenticate parameter to enable or disable (the default) mutual authentication.

   Protocol requirements mandate that the server present its certificate to the client application. Optionally, the server can implement mutual authentication by requesting a certificate from the client application, and authenticating the certificate offered by the client.

Upon receiving a server certificate request, the client application must respond with a certificate; failure to do so results in authentication failure.

```
ACMEPACKET(tls-profile)# mutual-authenticate disabled
ACMEPACKET(tls-profile)#
```

8. Retain the default value, compatibility, for the tls-version parameter.
9. Retain default values for all other parameters.
10. Use done, exit, and verify-config to complete tls-profile configuration.
11. Repeat Steps 1 through 10 to configure additional tls-profiles as required.

Palladion Probe Enhancement

Performance enhancements were made to the Palladion Probe functionality. Release S-C[xz]6.3.9M1 simplifies the operation of software-based Palladion probes by enabling the transmission of IPFIX data to one or more Palladion Mediation Engines, possibly on different sub-nets. This enhancement requires a slight change in the ACLI hierarchy -- specifically, the removal of the network-interface parameter from the comm-monitor configuration object, and its transfer to the monitor-collector configuration object.

Consequently, users who are migrating from a previous S-C[xz]6.3.9 release to S-C[xz]6.3.9M1 must be aware of the following anomaly. After the upgrade, probes based/anchored on media interfaces revert to the default network-interface value of wancom0:0.

The following illustrates a pre-S-C[xz]6.3.9M1 configuration.

```
comm-monitor
       state                     enabled
       qos-enable                disabled
       sbc-grp-id                0
       tls-profile
       network-interface         M10:0
       monitor-collector
              address            172.16.29.102
              port               4739
```

The following illustrates the upgraded S-C[xz]6.3.9M1 configuration.

```
comm-monitor
       state                     enabled
       qos-enable                disabled
       sbc-grp-id                0
       tls-profile
       monitor-collector
              address            172.16.29.102
              port               4739
              network-interface  wancom0:0
```

Note: Restoration of prior service requires a simple workaround, namely, the update of the network-interface parameter to its original value of M10:0.

22

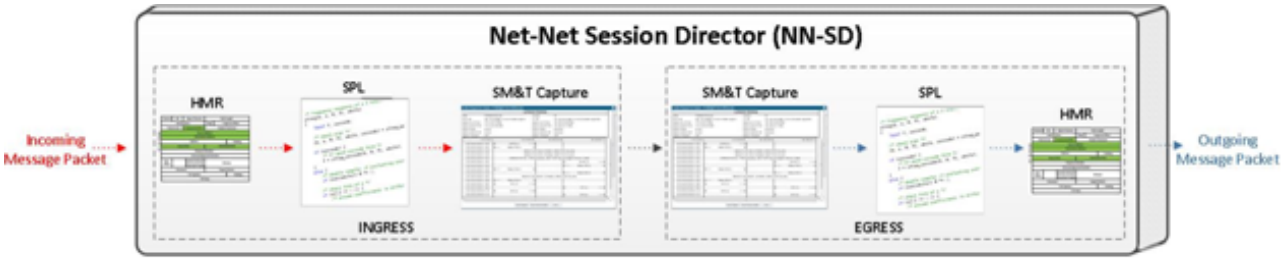
SIP Monitor & Trace

Introduction

The SIP Monitor and Trace feature provides the ability to set filters on the Net-Net ESD for filtering SIP session data, and displaying the results in a Web-based graphical user interface (GUI). You can use the data for the purpose of troubleshooting your Net-Net ESD(s).

This section provides information about how SIP Monitor and Trace works, configuring filters for filtering SIP session data, and how you can view/analyze monitored SIP session information via the GUI.

The SIP Monitor and Trace feature allows SIP sessions on the NN-SD(s) in your network to be monitored. The Net-Net SD captures SIP messages, applies the Header Manipulation Rules (HMR) configured on the Net-Net SD, and then applies the Session Plug-in Language (SPL) to that message. When the message is sent out from the Net-Net SD, it applies the SPL, then applies the HMR, and then sends out the captured SIP message.



HMR - Header Manipulation Rules
SPL- Session Plug-in Language
SM&T- SIP Monitor and Trace

You configure the monitoring process to use filters to filter the active session data from original ingress or final egress SIP session messages. These filters are based on Acme Packet's Command Line Interface (ACLI)-configured filters matching criteria or dynamic events that occur, and are used for the purpose of troubleshooting SIP sessions on the network.

As a system monitors active sessions, it captures data using the following filters:

- Static filters—Configured filters that filter the data on ingress and egress requests and responses in the SIP session dialogs.

  You configure these filters via the ACLI as part of the Net-Net ESD configuration. The configured filters save to the current configuration (after saving and activating the configuration).

- Dynamic filters—Filters you specify that match information in the ingress/egress SIP messages according to the filters you dynamically specified.

  You use the ACLI to specify these filters, but there is no change to the current configuration. The filters take effect immediately, and do not require the use of the "Save" and Activate commands. Using dynamic filters is recommended if you want to set specific filters but make no changes to the current configuration.

  For more information about configuring static filters and dynamic filters, see *Filters to Configure* and *Dynamic Filters*

When a filter configuration is enabled, the system matches the values in the configured filters to the headers of messages before it applies any changes. If no match is found in the headers during monitoring, the system uses the filter defaults in the system configuration to perform the filtering. The system logs the filter results along with any additional call details and displays the results in the GUI.

The following illustration shows the SIP Monitor and Trace flow process.

Filters to Configure

This section provides information about enabling the use of SIP Monitor and Trace filters you can configure on the Net-Net ESD. It includes a description and examples of the filter objects and attributes you can set to monitor specific SIP session data on the Net-Net ESD.

Filter Objects

The Net-Net ESD provides configuration objects you can set on the Net-Net ESD to customize filters for SIP Monitor and Trace. The system can monitor and filter specific SIP session data and display it to the GUI. The filter objects you can configure include:

Filters	Description
filter-config	Object that allows you to create custom filters to use for SIP Monitor and Trace. You can then configure session agents (SA) and/or realms to use these filters, or set sip-monitoring to use the filters on a global basis. For more information, see *Creating Custom Filters* .
sip-monitoring	Object that allows you to configure SIP Monitor and Trace features. Note: You must configure the sip-monitoring object to enable filtering. A session agent and/or realm must also be configured, or you must set filtering on a global basis, for Monitor and Trace to occur.
state	Attribute that enables/disables SIP Monitor and Trace. For more information, see *Enabling Disabling SIP Monitoring & Tracing* .
monitoring-filters	Attribute that allows you to specify the name of the custom filter(s) to use on a global basis. This value is based on the filter(s) created in "filter-config". You can also specify an * (asterisk) as a value for this attribute, which monitors all session data on the Net-Net ESD. For more information, see *Using Filters to Monitor on a Global-Basis* .
interesting-events	Object that allows you to configure the following attributes:

Filters	Description
	type - Sets the interesting events to monitor (short-session, local-rejection) trigger-threshold - Sets the number of interesting events that must occur within the time set by the trigger-window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started. trigger-timeout - Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. For example, if trigger-timeout is set to 40, and no interesting events occur in 40 seconds, then monitoring never starts. Note: Interesting Events are always enabled on a global-basis on the Net-Net ESD. For more information, see Configuring Interesting Events .
trigger-window	Attribute to specify the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. For example, if type is set to short-session, trigger-threshold is set to 2, and trigger-window is set to 60, monitoring begins when the Net-Net ESD has discovered 2 short-session events in a 60 second window. For more information, see *Configuring a Trigger Window* .

The following paragraphs provide information and procedures for configuring these features.

Creating Custom Filters

You can create single or multiple custom, session filters on the Net-Net ESD for Monitor and Trace purposes. These filters allow incoming and outgoing session data to be filtered with specific information and then displayed to the GUI. You can use the custom filter(s) during monitoring on a global basis, or when monitoring session agent (SAs) and/or realms.

You create custom filters using the filter-config object at the path **Configure terminal** > **session-router** > **filter-config** in the ACLI.

The following table identifies the attributes you can configure for each filter.

Filters	Description
filter-config	Object that allows you to create a custom filter(s) to be used for Monitor and Trace on the Net-Net ESD.
name	Name of the custom filter. Note: You specify this filter name when configuring global monitoring, SA monitoring, and/or realm monitoring.
address	IP address to be filtered. Depending on the value you specify for this attribute, filtering matches the IP address or IP address and netmask, in the message header. For example: 1.1.1.1 is <IP address> 1.1.1.1/24 is <IP address>/<Netmask>
user	Phone number or user-part to be filtered. Depending on the value you specify for this attribute, filtering matches the phone number string or the user-part with the following header information if it exists in the message:

Filters	Description
	From URI, To URI, Request URI, P-Preferred URI, P-Asserted Identity, P-Associated URI, P-Called Party URI.

You can define a single or multiple filters with specific names and then specify the filter name(s) to use for global monitoring, session agent monitoring, and/or realm monitoring.

Creating a Custom Filter

Use the following procedure to create a custom filter on the Net-Net ESD.

To configure a filter(s):

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type filter-config and press Enter to access the filter configuration-related attributes.

```
ACMEPACKET(session-router)# filter-config
ACMEPACKET(filter-config)#
```

   name—Enter a name to assign to this filter. Valid values are alpha-numeric characters. Default is blank.

```
ACMEPACKET(filter-config)# name FILTER1
```

   Note: You can use this filter name when configuring monitoring on a global-basis, or when monitoring session-agents and/or realms.

   address—Enter the IP address to apply to this filter. You can specify netmask if required. IP Address must be entered in dotted decimal format (0.0.0.0). Default is 0.0.0.0.

```
ACMEPACKET(filter-config)# address 1.1.1.1    (filters on IP address)
ACMEPACKET(filter-config)# address 1.1.1.1/24 (filters on IP address and
netmask)
```

   user—Enter a phone number or user-part to apply to this filter. Valid values are numeric characters. Default is blank.

```
ACMEPACKET(filter-config)# user 5551212
```

   You must specify either the phone number OR user part for the user attribute. If you want both the phone number AND user part to be filtered, you must create separate filters to set each value.

4. Enter done to save the filter.

```
ACMEPACKET(filter-config)# done
```

5. Enter exit to exit the filter configuration.

```
ACMEPACKET(filter-config)# exit
```

6. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

7. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

8. Enter save-config to save the filter configuration.

```
ACMEPACKET# save-config
```

9. Enter activate-config to activate the filter configuration.

```
ACMEPACKET# activate-config
```

Multiple Custom Filter Examples

The following examples show three custom filters (FILTER1, FILTER2, and FILTER3) created for SIP Monitor and Trace on the Net-Net ESD.

- Filter 1

```
ACMEPACKET(filter-config)# name FILTER1
ACMEPACKET(filter-config)# address 1.1.1.1
ACMEPACKET(filter-config)# user 5551212
```

- Filter 2

```
ACMEPACKET(filter-config)# name FILTER2
ACMEPACKET(filter-config)# address 3.3.3.3/24
ACMEPACKET(filter-config)# user 1781
```

- Filter 3

```
ACMEPACKET(filter-config)# name FILTER3
ACMEPACKET(filter-config)# user sip
```

You can specify the Net-Net ESD monitoring process to use FILTER1, FILTER2, and/or FILTER3 for global monitoring, or for monitoring SAs and/or realms. However, before you apply the custom filters, you can enable/disable SIP monitoring on the Net-Net ESD.

To enable/disable SIP monitoring, see *Enabling Disabling SIP Monitoring & Tracing*. To use a custom filter(s) on a global basis, see *Using Filters to Monitor on a Global-Basis*. To use a custom filter(s) when monitoring SAs, see *Using Filters when Monitoring Session Agents*. To use a custom filter(s) when monitoring realms, see *Using Filters when Monitoring Realms*.

Enabling Disabling SIP Monitoring & Tracing

You can enable or disable the Net-Net ESD to perform SIP monitoring using the state parameter at the path **Configure terminal** > **session-router** > **sip-monitoring** in the ACLI.

Use the following procedure to enable/disable SIP monitoring on the Net-Net ESD.

Note: You must enable the sip-monitoring object for monitoring and filtering to occur on the Net-Net ESD. With sip-monitoring enabled, you can configure a filter(s) on a global basis, as well as for a session agent and/or a realm. You can also initiate dynamic filter commands.

To enable/disable sip-monitoring:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-monitoring and press Enter to access the SIP monitoring-related attributes.

```
ACMEPACKET(session-router)# sip-monitoring
ACMEPACKET(sip-monitoring)#
```

   state—Enter whether or not to enable the sip monitoring on the Net-Net ESD. Default is enabled. Valid values are:

   - enabled (default)
   - disabled

4. Enter done to save the setting.

```
ACMEPACKET(sip-monitoring)# done
```

5. Enter exit to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

6. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

7. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

8. Enter save-config to save the filters.

```
ACMEPACKET# save-config
```

9. Enter activate-config to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

10. Configure global filters, or assign filters to a session agent and/or realm. For more information, see the following:

- *Using Filters to Monitor on a Global-Basis*
- *Using Filters when Monitoring Session Agents*
- *Using Filters when Monitoring Realms*

With sip-monitoring enabled, you can also initiate dynamic filter commands if required. For more information about dynamic filter commands, see *Dynamic Filter Commands*.

Using Filters to Monitor on a Global-Basis

The Net-Net ESD allows you to filter SIP session data on a global-basis using the monitoring-filters object at the path **Configure terminal** > **session-router** > **sip-monitoring** > **monitoring-filters** in the ACLI. You can apply a single or multiple custom filter for global monitoring. For more information about creating a custom filter, see *Creating a Custom Filter*.

Note: For SIP Monitor and Trace to trigger interesting-events, a filter value must be configured for the monitoring-filters object.

To configure filtering on a global basis:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-monitoring and press Enter to access the SIP monitoring-related attributes.

```
ACMEPACKET(session-router)# sip-monitoring
ACMEPACKET(sip-monitoring)#
```

4. Type select and press Enter to select the sip-monitoring objects.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the "+" before the filter name you are adding. Use a "-" to remove filter names. Enter an * (asterisk) to filter all session data.

```
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(sip-monitoring)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the Net-Net ESD uses the * to filter all session data.

5. Enter done to save the configuration.

```
ACMEPACKET(sip-monitoring)# done
```

6. Enter exit to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

7. Enter done to save the sip-monitoring configuration.

```
ACMEPACKET(session-router)# done
```

8. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

9. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter activate-config to activate the configuration.

```
ACMEPACKET# activate-config
```

Using Filters when Monitoring Session Agents

You can configure the Net-Net ESD to perform filtering of SIP session data for session agent (SA) configurations. You must specify the hostname of the SA and the filter to use to perform the filtering, at the path **Configure terminal** > **session-router** > **session-agent**in the ACLI. For more information about creating a custom filter, see *Creating a Custom Filter*.

To configure filtering for a Session Agent:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter to access the session agent-related attributes.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Type select and press Enter.

```
ACMEPACKET(session-agent)# select
ACMEPACKET(session-agent)#
```

hostname—Specify the hostname of the session agent to which you want to apply the custom filter(s).

```
ACMEPACKET(session-agent)# hostname SA1
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the “+” before the filter name you are adding. Use a “-” to remove filter names. Enter an * (asterisk) to filter all SIP session data.

```
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(session-agent)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the Net-Net ESD uses the * to filter all session data.

5. Enter done to save the configuration.

```
ACMEPACKET(session-agent)# done
```

6. Enter exit to exit the session-agent configuration.

```
ACMEPACKET(session-agent)# exit
```

7. Enter done to save the configuration.

```
ACMEPACKET(session-router)# done
```

8. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

9. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter activate-config to activate the configuration.

```
ACMEPACKET# activate-config
```

Using Filters when Monitoring Realms

You can configure the Net-Net ESD to perform filtering of SIP session data for realm configurations. You must specify the realm identifier and the filter to use to perform the filtering, at the path Configure terminal->media-manager->realm-config in the ACLI. For more information about creating a custom filter, see *Creating a Custom Filter*.

To configure filtering for a realm:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter to access the media manager-related objects.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter to access the realm configuration-related attributes.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Type select and press Enter.

```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)#
```

identifier—Specify the identifier of the realm to which you want to apply the custom filter(s).

```
ACMEPACKET(realm-config)# identifier REALM1
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the "+" before the filter name you are adding. Use a "-" to remove filter names. Enter an * (asterisk) to filter all SIP session data.

```
ACMEPACKET(realm-config)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(realm-configg)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(realm-config)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(realm-config)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the Net-Net ESD uses the * to filter all session data.

5. Enter done to save the configuration.

```
ACMEPACKET(realm-config)# done
```

6. Enter exit to exit the realm-config configuration.

```
ACMEPACKET(realm-config)# exit
```

7. Enter done to save the configuration.

```
ACMEPACKET(media-manager)# done
```

8. Enter exit to exit the media-manager configuration.

```
ACMEPACKET(media-manager)# exit
```

9. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter activate-config to activate the configuration.

```
ACMEPACKET# activate-config
```

Global SA and Realm Filter Examples

The following are examples of global, session agent, and realm filters configured on the Net-Net ESD. These examples assume that FILTER1, FILTER2, and FILTER3 have been pre-configured as custom filters.

Global Filter

```
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER3
```

This filter captures the SIP session data based on the filter settings in FILTER1 and FILTER3 only, for all sessions on the Net-Net ESD.

Session Agent Filters

```
ACMEPACKET(session-agent)# hostname SA1
ACMEPACKET(session-agent)# monitoring-filters FILTER2
ACMEPACKET(session-agent)# hostname SA2
ACMEPACKET(session-agent)# monitoring-filters FILTER2,FILTER3
```

These filters capture the SIP session data for SA1 only, based on the filter settings in FILTER2, and the SIP session data for SA2 only, based on the filter settings in FILTER2 and FILTER3.

Realm Filters

```
ACMEPACKET(realm-config)# identifier REALM1
ACMEPACKET(realm-config)# monitoring-filters *
ACMEPACKET(realm-config)# identifier REALM2
ACMEPACKET(realm-config)# monitoring-filters FILTER1
```

These filters capture all SIP session data for REALM1, and the SIP session data for REALM2 only, based on the filter settings in FILTER1.

Note: If you leave a monitoring-filter field blank, no monitoring takes place for that object.

Interesting Events

Interesting events on the Net-Net ESD are those events that are considered “interesting” for the purpose of troubleshooting SIP sessions in your network. You can specify the type of interesting event you want to filter using the object, interesting-events at the path, **Configure terminal** > **session-router** > **sip-monitoring** > **interesting-events** in the ACLI.

Currently, there are two types of interesting events that the Net-Net ESD can monitor:

- short-session (short session events on the Net-Net ESD)
- local-rejection (local rejection events on the Net-Net ESD)

You can use the following trigger attributes to specify time provisioning for the interesting events:

- trigger-threshold
- trigger-timeout

Note: You can also set a trigger-window object to support these trigger attributes. For more information, see *Configuring a Trigger Window*.

The following table identifies the attributes you can set for the interesting-events object.

Filter	Description
interesting-events	Allows you to configure trigger attributes that apply to the filters you set on the Net-Net ESD. You can configure the following interesting-event attributes: Note: Interesting Events are always enabled on a global-basis on the Net-Net ESD.
type	Sets the interesting events to monitor (short-session, local-rejection)
trigger-threshold	Sets the number of interesting events that must occur within the time set by the trigger-window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started.
trigger-timeout	Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. For example, if trigger-timeout is set to 40, and no interesting events occur in 40 seconds, then monitoring never starts.

The Net-Net ESD considers short session and local rejection interesting events. A session is viewed as a short session if the length of time, in seconds, is equal to or below the short-session-duration value configured at the path **Configure terminal** > **session-router** > **session-router-config** > **short-session-duration**. A local rejection can occur when sessions are locally rejected at the Net-Net ESD for any reason (for example, Session Agent (SA) unavailable, no route found, SIP signaling error, etc.)

If a short session or local rejection event occurs, the Net-Net ESD uses the values configured for the trigger attributes to determine when to start filtering the SIP session data.

If a short session event occurs when the Net-Net ESD is NOT monitoring, the event information is taken from the last BYE that occurred in the session; therefore, only some parts of the call flow may display in the GUI. If a local rejection event occurs when the Net-Net ESD is NOT monitoring, it displays only the information in the last rejected transaction.

Use the following procedure to configure interesting events for SIP Monitor and Trace on the Net-Net ESD.

Interesting Events Configuration

To configure interesting events:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-router again and press Enter to access the session router configuration-related attributes.

```
ACMEPACKET(session-router)# session-router
ACMEPACKET(session-router-config)#
```

short-session-duration—Enter the maximum session duration, in seconds, to be considered a short session. Default is 0 (disabled). Valid values are 0 to 999999999.

```
ACMEPACKET(session-router-config)# short-session-duration 30
```

4. Enter done to save the filters.

```
ACMEPACKET(session-router-config)# done
ACMEPACKET(session-router-config)#
```

5. Enter exit to exit the interesting-events configuration.

```
ACMEPACKET(session-router-config)# exit
ACMEPACKET(session-router)#
```

6. Type sip-monitoring and press Enter to access the SIP monitoring-related attributes.

```
ACMEPACKET(session-router-config)# sip-monitoring
ACMEPACKET(sip-monitoring)#
```

7. Type select and press Enter.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

8. Type interesting-events and press Enter to access the interesting events-related attributes.

```
ACMEPACKET(sip-monitoring)# interesting-events
ACMEPACKET(interesting-events)#
```

type—Enter the type of interesting event you for which you want to filter. Default is blank and disables this filter. Valid values are:

- short-session
- local-rejection

```
ACMEPACKET(interesting-events)# type short-session
```

trigger-threshold — (optional) Enter the number of interesting events that must occur within the time set by the trigger window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started.Default is 0 (disabled). Valid values are 0 to 999999999.

```
ACMEPACKET(interesting-events)# trigger-threshold 50
```

trigger-timeout —Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. Default is 0 (trigger always on). Valid values are 0 to 999999999.

```
ACMEPACKET(interesting-events)# trigger-timeout 30
```

9. Enter done to save the filters.

```
ACMEPACKET(interesting-events)# done
```

10. Enter exit to exit the interesting-events configuration.

```
ACMEPACKET(interesting-events)# exit
```

11. Enter exit to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

12. Enter done to save the configuration.

```
ACMEPACKET(session-router)# done
```

13. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

14. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

15. Enter save-config to save the filters.

```
ACMEPACKET# save-config
```

16. Enter activate-config to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

Configuring a Trigger Window

The trigger-window attribute specifies the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. For example, if “interesting-event” type is set to short-session, “trigger-threshold” is set to 2, and trigger-window is set to 60, monitoring begins when the Net-Net ESD has discovered 2 short-session events in a 60 second window.

Use the following procedure to configure a trigger window.

To configure a trigger window:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-monitoring and press Enter to access the SIP monitoring-related attributes.

```
ACMEPACKET(session-router)# sip-monitoring
ACMEPACKET(sip-monitoring)#
```

4. Type select and press Enter.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

   trigger-window—Enter the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. Default is 30. Valid values are 0 to 999999999. Zero (0) disables this the trigger-window parameter.

```
ACMEPACKET(sip-monitoring)# trigger-window 50
```

5. Enter done to save the filters.

```
ACMEPACKET(sip-monitoring)# done
```

6. Enter exit to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

7. Enter done to save the configuration.

```
ACMEPACKET(session-router)# done
```

8. Enter exit to exit the session-router configuration.

```
ACMEPACKET(session-router)# exit
```

9. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter save-config to save the filters.

```
ACMEPACKET# save-config
```

11. Enter activate-config to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

Example

The following is an example filter configuration, filtering interesting events with a trigger window on the Net-Net ESD. These parameters perform filtering on a global basis.

Monitoring Enabled on a Global Basis

```
ACMEPACKET(sip-monitoring)# state enabled
```

Short-Session Configured

```
ACMEPACKET(interesting-events)# type short-session
ACMEPACKET(interesting-events)# trigger-threshold 2
ACMEPACKET(interesting-events)# trigger-timeout 60
```

Local-Rejection Configured

```
ACMEPACKET(interesting-events)# type local-rejection
ACMEPACKET(interesting-events)# trigger-threshold 1
ACMEPACKET(interesting-events)# trigger-timeout 0
```

Trigger-Window Configured

```
ACMEPACKET(sip-monitoring)# trigger-window 120
```

The configuration above has global SIP monitoring enabled and is set to capture interesting events that are short-session and local-rejection events.

Per the triggers for the short-session configuration, if 2 (trigger-threshold) short-session events occur in a window of 120 seconds (trigger-window), then monitoring is started. If no short-session events occur after 60 seconds (trigger-timeout), no monitoring is started.

Per the triggers for the local-rejection configuration, if more that 1 (trigger-threshold) local-rejection event occurs in a window of 120 seconds (trigger-window), then monitoring is started. The value of 0 (trigger-timeout) indicates that monitoring is always enabled for this event.

Dynamic Filters

The SIP Monitor and Trace feature provides a time-saving feature of adding filters dynamically, and turning the filters ON and OFF as required. The filtering process performs on a dynamic basis dependant on the filters you specify.

Dynamic Filter Commands

You can use the ACLI to initiate the following dynamic filtering commands:

- capture start—starts the filters you specify in the filter syntax
- capture stop—stops the filters you specify in the filter syntax

Note: Initiating these commands does not change the values set in the ACLI-configured filters on the Net-Net ESD. The Net-Net ESD uses the dynamic filters until you initiate a stop command.

The syntax for the dynamic filter commands are:

capture start <main filter> <subfilter(s)>

capture stop <main filter> <subfilter(s)>

You MUST enter a <main filter> and a <subfilter(s)> when initiating the “capture start” and capture stop commands.

The following table identifies the values you can use for each attribute in the command syntax.

Syntax Attribute	Values
<main filter>	global - monitors and captures all realm <realm name> - monitors and captures everything matching realm session-agent <session-agent name> - monitors and captures everything matching session agent. int-ev <short-session | local-rejection> - monitors and captures everything matching a short- session and/or local-rejection.
[<subfilter(s)>]	* - monitors and captures all sessions.

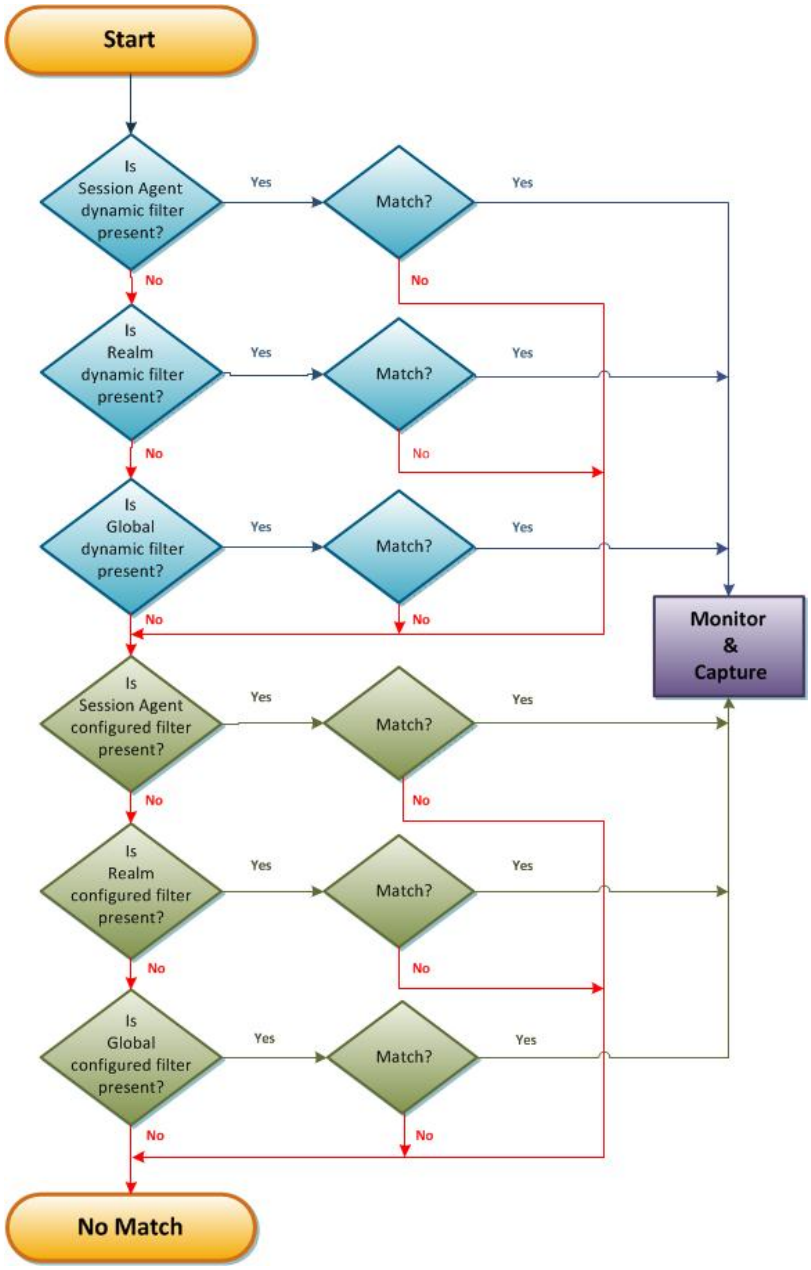
Syntax Attribute	Values
	user <Phone Number or User Part URI> - monitors and captures everything that matches this phone number or user part. addr-prefix <IP address or IP address and netmask> - monitors and captures everything that matches this address or address prefix.

Examples

The following table provides examples for using the dynamic filter commands.

Example	Description
capture start global *	Captures all session data.
capture start global user USER1	Captures all session data for USER1.
capture start global addr-prefix 1.1.1.1	Captures all session data for IP address 1.1.1.1.
capture start global addr-prefix 1.1.1.1/24	Captures all session data for IP address 1.1.1.1 using netmask of 24.
capture start session-agent 172.1.1.1 addr-prefix 10.10.10.10	Captures session data for SA 172.1.1.1 at IP address 10.10.10.10.
capture start int-ev local-rejection	Captures session data for interesting events that occur that are of type local-rejection.
capture start int-ev short session	Captures session data for interesting events that occur that are of type short-session.

The following flow chart shows the dynamic filter process.



Note: Dynamic filters are only removed after a reboot/switchover of the Oracle Enterprise Session Border Controller.

Issuing another dynamic command may or may not affect previous dynamic commands that were already initiated. If you issue a dynamic command with a <main filter> object, and then issue another command with the same <main filter> object, the new command tasks precedence. If you issue a dynamic command with a different <main filter> object, then the Oracle Enterprise Session Border Controller uses both <main filter> commands to monitor traffic.

For example, if you enter the following dynamic command:

```
ACMEPACKET# capture start realm1 user 123
```

and then enter:

```
ACMEPACKET# capture start realm2 user 123
```

The Oracle Enterprise Session Border Controller monitors realm1 AND realm2 with user 123.

To stop dynamic filter commands, you can initiate the capture stop <main filter> command. For example:

```
ACMEPACKET# capture stop realm1 user 123
```

To stop configured filters, you must manually remove them from the ACLI configuration.

Clearing all Dynamic Filters

You can clear all dynamic filters using the following command:

- reset monitoring dynamic-commands—clears all dynamic filters previously initiated

The Net-Net ESD maintains a record of all dynamically initiated active filters. When you initiate this reset command, the Net-Net ESD searches through all of the filters and resets all the dynamic filters for each main filter (realm, session-agent, session-group, interesting event).

Example

The following command is an example of using the reset command to clear all dynamic capture filters.

```
ACMEPACKET# reset monitoring dynamic-commands
```

The following message displays: Reset all dynamically created monitoring capture commands....

Clearing Event Monitoring Records

You can clear all records stored in the event monitoring in-memory database using the following command:

- reset monitoring records—clears all event monitoring records from the in-memory database.

Use the following procedure to clear all event monitoring records.

To clear event monitoring records:

1. At the prompt, type reset monitoring records, and press Enter.

   ```
   ACMEPACKET# reset monitoring records
   ```

   The following prompt displays:

   ```
   All in-memory event monitoring records will be deleted [y/n]?:
   ```

2. Type y and press Enter.

   ```
   All in-memory event monitoring records will be deleted [y/n]?: y
   ```

   The following message displays: Deleting the in-memory event records.

   If you enter n for Step 2, the following message displays: Cancelling the reset.

   No event monitoring records are deleted.

23

Personal Profile Manager (PPM) Proxy

Introduction

The Net-Net ESD includes a Personal Profile Manager (PPM) proxy feature. PPM is a web service that runs as part of Avaya Aura Session Manager and Aura System Manager. Local and remote SIP clients may download configuration data from the PPM proxy using SOAP messages over HTTP or HTTPS, enabling soft keys to be customized and contact lists to be loaded. Unfortunately, in enterprise networks certain messages may refer to private IP addresses, which are not routable from remote clients. Acme Packet now incorporates an application proxy in the Net-Net ESD for such messages, replacing the internal IP addresses with the Net-Net ESD's external SIP interface IP address.

The PPM proxy supports incoming messages over HTTP and HTTPS on a configurable IP address / port. If using HTTPS, the PPM proxy uses a selectable server certificate for Transport Layer Security (TLS).

Remote clients accessing the PPM proxy are authenticated by HTTP digest authentication, using their SIP credentials. The PPM proxy forwards such challenges and responses transparently to the PPM web service for which it is configured.

Since the PPM proxy could potentially be a target of a denial-of-service (DoS) attack, the Net-Net ESD allows you to set DoS rules to protect the proxy port as part of standard configurations. For configuring DoS on the Net-Net ESD, see the chapter on security.

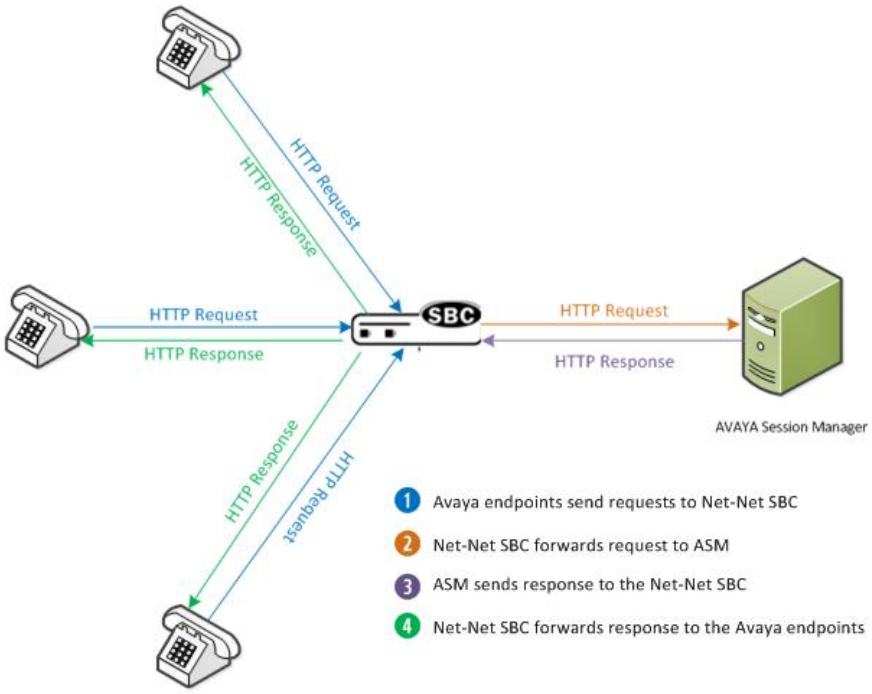
Net-Net ESD as ALG for HTTP HTTPS

The Net-Net ESD functions as an HTTP Application Layer Gateway (ALG) for HTTP/HTTPS traffic that originates on Avaya endpoints and terminates on the Avaya Session Manager (ASM) as follows:

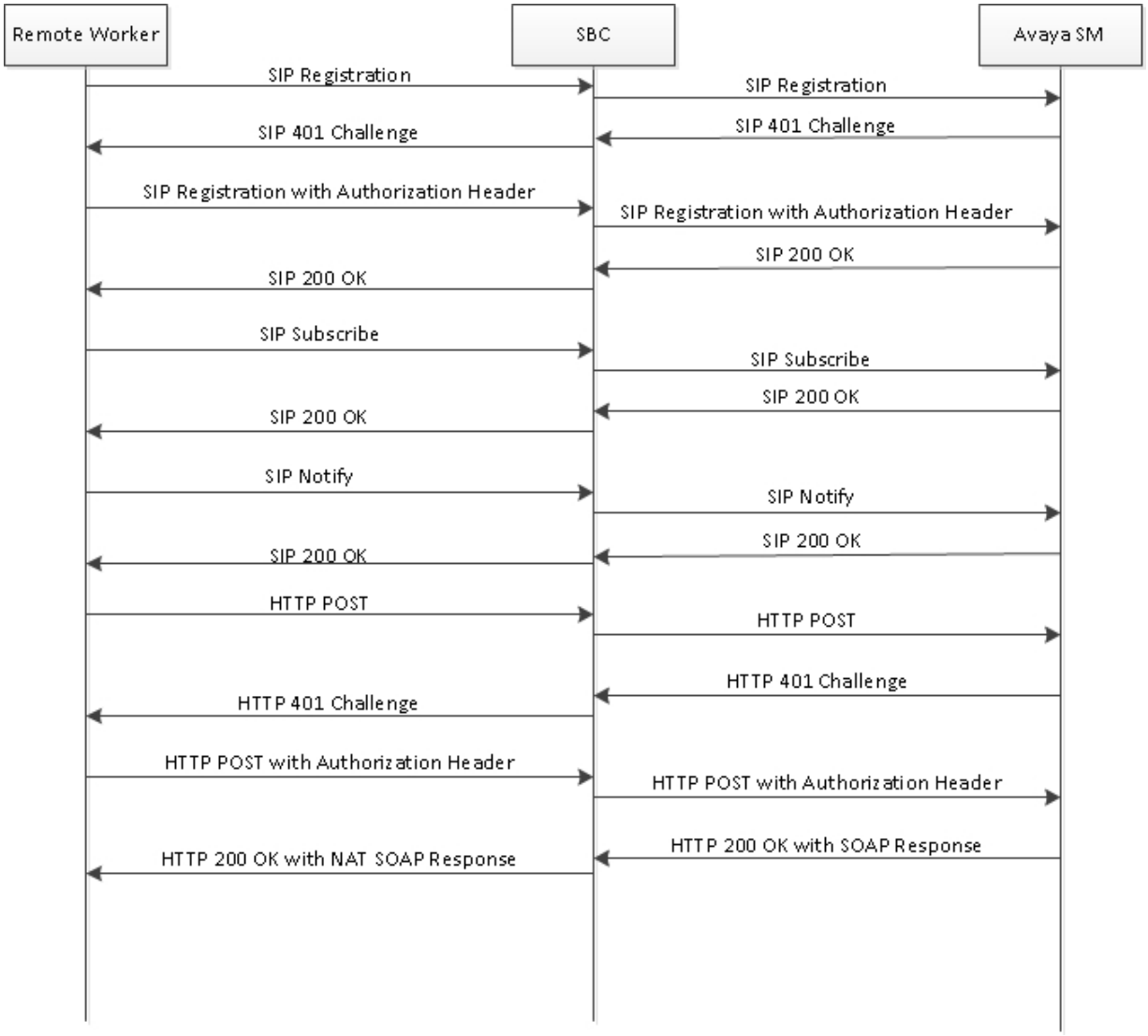
1. The Net-Net ESD receives HTTP requests from Avaya endpoints on a user configurable IP address and port.
2. The Net-Net ESD then forwards the requests to a user configurable destination which is the IP address and port of the ASM.
3. The response to the HTTP request is sent from the ASM to the Net-Net ESD.

   The Net-Net ESD parses the HTTP response and searches for getHomeServerResponse and getHomeCapabilitiesResponse messages. If the getHomeServerResponse message is found, the Net-Net ESD replaces any text between the <PpmServer> or <SipServer> tags with the IP address of the public interface on which the HTTP-ALG is configured. If the getHomeCapabilitesResponse is found, the Net-Net ESD replaces any text contained between the <ServiceURI> tags with the IP address of the public interface on which the HTTP-ALG is configured.

4. After the Net-Net ESD is done processing the response, it forwards the response to the originating Avaya endpoint. The following illustration shows how the Net-Net ESD sends/receives HTTP requests/responses to the Avaya Session Manager.



The following is the call flow that occurs as the HTTP/HTTPS requests and responses are passed between the Avaya endpoints, the Net-Net ESD, and the ASM.



Configuring the PPM Proxy on the Net-Net ESD

To configure the PPM proxy on the Net-Net ESD, you use the http-alg object under session-router, and the http-alg-private or http-alg-public settings. Use the following procedure to configure the PPM proxy on the Net-Net ESD.

To configure the PPM proxy:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type http-alg and press Enter.

```
ACMEPACKET(session-router)# http-alg
ACMEPACKET(http-alg)#
```

4. name—Enter the name (unique identifier) of the HTTP proxy. Valid values are alpha-numeric characters. Default is blank.
5. state—Enter the operational status of the HTTP proxy. Valid values are:

- enabled - (default) Enables the HTTP proxy.
- disabled - Disables the HTTP proxy.

6. description—Enter a description of the HTTP proxy. Valid values are alpha-numeric characters. Default is blank.
7. private—Allows you to configure a private/core-side interface (inside the network) for forwarding the incoming HTTP SOAP Requests received from the public side.
8. public—Allows you to configure a public-side interface (outside the network) to receive incoming HTTP SOAP Requests from the remote worker.

Private Settings on the Net-Net ESD

To set a private setting on the Net-Net ESD:

1. Type http-alg-private and press Enter.

```
ACMEPACKET(http-alg)# http-alg-private
ACMEPACKET(http-alg-private)#
```

   The private /core side is used to communicate with the Avaya Session Manager (ASM) and forward the incoming HTTP SOAP Requests received from the public side (from outside the network). You define the IP address, port, and TLS certificate used in establishing communication with the ASM by setting this http-alg-private attribute.
2. realm-id—Name of the realm that the Net-Net ESD uses to proxy the HTTP request. Valid values are alpha-numeric characters. Default is blank.
3. address—IPv4 or IPv6 IP address from which the Net-Net ESD forwards the incoming HTTP request. Valid values must be in the format of 0.0.0.0. Default is blank.
4. destination-address—IPv4 or IPv6 IP address of the destination server to which the HTTP request is forwarded. Valid values must be in the format of 0.0.0.0. Default is blank.
5. destination-port—Port on which the destination server is listening for HTTP traffic. Valid values are 1 to 65535. Default is 80.
6. tls-profile—The TLS profile used to establish a secure connection with the destination server. Setting this attribute enables HTTP proxy to listen for HTTPS traffic. Valid values are alpha-numeric characters. Default is blank.
7. Type done and press Enter.

```
ACMEPACKET(http-alg-private)# done
ACMEPACKET(http-alg-private)#
```

8. Type exit and press Enter.

```
ACMEPACKET(http-alg-private)# exit
ACMEPACKET(http-alg)#
```

9. Type exit and press Enter.

```
ACMEPACKET(http-alg)# exit
ACMEPACKET(session-router)#
```

10. Save the configuration.

Public Settings on the Net-Net ESD

To set a public setting on the Net-Net ESD:

1. Type http-alg-public and press Enter.

```
ACMEPACKET(http-alg)# http-alg-public
ACMEPACKET(http-alg-public)#
```

   The public side (outside the network) is used to receive incoming HTTP SOAP Requests from the remote worker. You define the IP address, port, and TLS certificate used to establish a connection with the remote worker by setting this http-alg-public attribute.

2. realm-id—Name of the realm that the Net-Net ESD uses to listen for the HTTP request. Valid values are alpha-numeric characters. Default is blank.
3. address—IPv4 or IPv6 IP address on which the Net-Net ESD is listening for HTTP traffic. Valid values must be in the format of 0.0.0.0. Default is blank.
4. port—Port on which the Net-Net ESD is listening for HTTP traffic. Valid values are 1 to 65535. Default is 80.
5. tls-profile—The TLS profile used to establish a secure connection with the remote worker. Setting this attribute enables HTTP proxy to listen for HTTPS traffic. Valid values are alpha-numeric characters. Default is blank.
6. Type done and press Enter.

```
ACMEPACKET(http-alg-public)# done
ACMEPACKET(http-alg-public)#
```

7. Type exit and press Enter.

```
ACMEPACKET(http-alg-public)# exit
ACMEPACKET(http-alg)#
```

8. Type exit and press Enter.

```
ACMEPACKET(http-alg)# exit
ACMEPACKET(session-router)#
```

9. Save the configuration.

PPM XML Mapping to ACLI Parameters

Each of the PPM parameters in the ACLI map to specific XML tags. The following table provides the XML/ACLI parameter mapping.

Parameter Name	XML Tag
http-alg	httpAlg
name	name
state	state
description	description
http-alg-private	httpAlgPrivate
realm-id	RealmID
address	address
destination-address	destination-address
destination-port	destination-port
tls-profile	tlsProfile
http-alg-public	httpAlgPublic
realm-id	RealmID
address	address
port	port
tls-profile	tlsProfile

Example PPM Proxy Configuration

The following is an example of a the PPM proxy configuration with private enabled.

```
session-router# show
    http-alg
        name                          Avaya
        state                         enabled
        description                   Avaya Proxy
        http-alg-private
            realm-id                   realmA
            address                    172.45.6.7
            destination-address        123.456.78.1
            destination-port           80
            tls-profile                tls1
        http-alg-public
            realm-id
            address
            port
            tls-profile
```

24

Remote Site Survivability

Release E-C[xz]6.4.0 M2 includes a new feature called Remote Site Survivability. This feature is the Oracle Enterprise Session Border Controller's ability of a Remote Office/Branch Office (ROBO) to detect the loss of communication over SIP-based telephony, to the Enterprise's core call processing Data Center. When loss of communication is detected over the SIP service, the ROBO Oracle Enterprise Session Border Controller dynamically switches into Survivable Mode, locally handling call processing and providing limited additional server functionality.

Note: Remote Site Survivability supports SIP only. It does not support the H.323.

The following are features of Remote Site Survivability:

- Works with or without High Availability (HA) operation.
- Configurable in real-time - no reboot required to enable this feature.
- Allows configuration of the feature via the Oracle Enterprise Session Border Controller Web GUI
- Maintains Historical Recording (HDR) statistics about being in survivability mode, such as:

  Whether or not the Oracle Enterprise Session Border Controller is in survivable mode using the ACLI command, show health.

  Length of time the Oracle Enterprise Session Border Controller was in survivable mode (records number of times and amount of time in survivability mode)

  Number of SIP messages handled in survivable mode

  Number of SIP users registered locally in survivable mode (both existing based on cache, and separately - new registrations).

How it Works

When configured for Survivability, the Oracle Enterprise Session Border Controller operates in either normal or survival mode. In normal mode, the IP wide area network (WAN) connection between the remote Oracle Enterprise Session Border Controller and the data center headquarters site is operational, and endpoints at the remote site register through the SBCs to an IP-PBX or Application Server (AS) at headquarters. Similarly, the Net-Net ESD forwards calls between endpoints to the IP-PBX or AS at headquarters. When an endpoint registers, the Oracle Enterprise Session Border Controller inserts a registration entry for the endpoint in its local registration cache.

When the IP connection to headquarters goes down, the Oracle Enterprise Session Border Controller operates in survival mode. In this mode, the system is able to detect any loss of connection (and subsequent re-connection) to the core data center based on a health score (For more information about health score, see *Survivability Health Score*). When it detects a loss of connection, it enters survival mode and locally processes registrations and session traffic

without routing them to the registrar. The Oracle Enterprise Session Border Controller also handles call routing in this mode. When a subsequent re-connection is detected, the system exits survival mode and proxies all registrations and session traffic once again to the data center (normal mode).

In "Survival Mode", the ROBO Oracle Enterprise Session Border Controller provides the following capabilities:

- Maintains SIP registrations for local SIP phones (based on existing registration cache).
- Provides local extension-to-extension calling and incoming public switched telephone network (PSTN), if available, to local extension dialing.
- Provides extension-to-PSTN calling through a media gateway (assuming a gateway is available) or alternatively, via a configured SIP trunk/route.
- Allows all new registration requests (without authentication) to be successful.
- Allows extensions to be dialed based on its multiple user identities (either identified by using P-Asserted-Identity or BroadSoft's proprietary mechanism). For more information about Survivability using a BroadSoft server, see *Remote Site Survivability with a BroadSoft Server*.

Survivability Health Score

When Survivability Mode is enabled on the Oracle Enterprise Session Border Controller, the system is able to detect any loss of connection (and subsequent re-connection) to the Enterprise's core data center based on a health score.

For the purpose of health monitoring, a sip-interface and one or more attached session agents can be logically grouped together by configuring a"service-tag" parameter to indicate the name of the session agent group. The service health score of the group is based upon the health status of the session agents within the group and can be configured using the session-agent-health parameter. The session-agent-health score can be a value between 0 and 100.

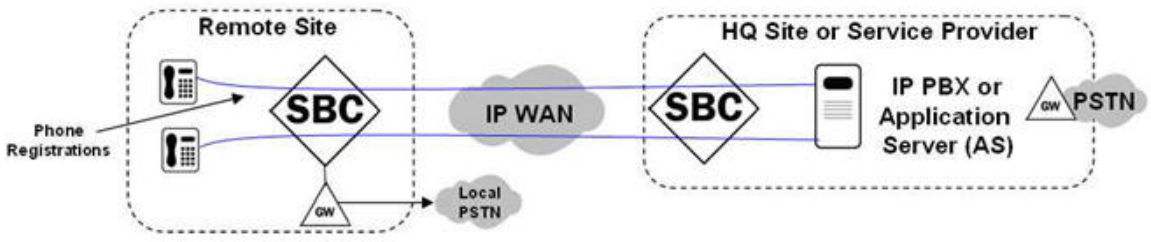
The determination of when to enter survival mode is determined by the session agent health score. The session-agent-health value is the amount that is deducted from the service health score when the session agent goes out of service. The sum of the service health values of all session agents assigned to a specific service tag must equal 100 to stay in normal mode. In cases where there is one session agent, the service health value is 100. For cases where there are two session agents, each session agent could have a service health of 50.

When the service health score goes down to zero the Oracle Enterprise Session Border Controller enters survival mode. While in survival mode, the Oracle Enterprise Session Border Controller continuously attempts to re-establish communications with the session agents. If communication is re-established, the Oracle Enterprise Session Border Controller adds the service agent health value of the session agent to the current service health score, and survival mode is exited if the service health score is above zero.

Note: For more information about configuring Survivability Mode and the Survivability health score, see *Configuring Remote Site Survivability using the ACLI* or *Configuring Remote Site Survivability using the Web GUI*.

Normal Behavior Call Process

The following illustration shows the normal call process behavior of the ROBO
Oracle Enterprise Session Border Controller connectivity to the Service Provider site (or headquarters site).

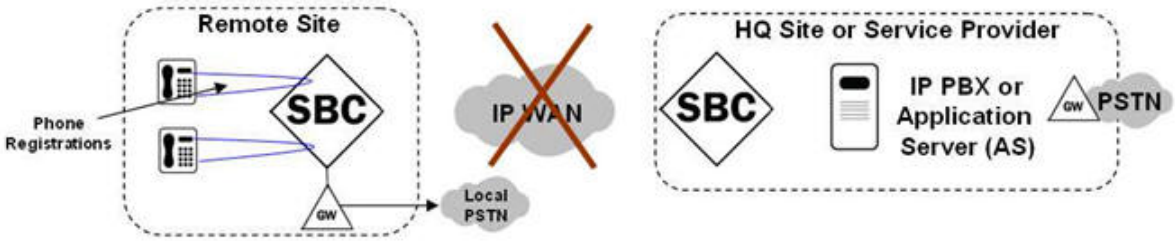


1. Phones register through the Oracle Enterprise Session Border Controller to the IP PBX or Application Server (AS) at the Headquarters or Service Provider site.

2. Phone-to-phone calls are proxied through the Oracle Enterprise Session Border Controllers to the IP PBX or AS at the Headquarters or Service Provider site.
3. Phone-to-Public Switched Telephone Network (PSTN) calls are routed to the Headquarters or Service Provider site, or sent out a local PSTN gateway.

Remote Survivable Call Process Behavior

The following illustration shows the remote survivable call process behavior of the ROBO Oracle Enterprise Session Border Controller when connectivity fails to the Service Provider site (or headquarters site).



1. Phones register directly on remote site Oracle Enterprise Session Border Controller.
2. Phone-to-phone calls are proxied directly on remote site Net-Net ESD.
3. Phone-to-PSTN calls are routed by remote site Oracle Enterprise Session Border Controller to local PSTN gateway.

Entering Survivable Mode

Registration Behavior

When the Oracle Enterprise Session Border Controller enters Survivable Mode, it performs as follows for registrations:

For endpoints already Registered...	For new Registration requests... (either new endpoints or endpoints whose registration expires when in Survivable Mode)
the Oracle Enterprise Session Border Controller acts as the registrar of the local SIP phones by providing 200 OK responses to subsequent REGISTER refresh messages from endpoints in the Oracle Enterprise Session Border Controller's reg-cache for the duration of Survivable mode. This presumes that "registration-caching" has been enabled in the Oracle Enterprise Session Border Controller onfiguration.	the Oracle Enterprise Session Border Controller allows the new Registrations to be successful (without providing Authentication), incorporating them into the Oracle Enterprise Session Border Controller registration cache.
the Oracle Enterprise Session Border Controller lowers the "reg-expires" value to 30 seconds by default for all Registration Requests between the endpoints and the Oracle Enterprise Session Border Controller.	the Oracle Enterprise Session Border Controller lowers the "reg-expires" value to 30 seconds by default for all Registration Requests between the endpoints and the Oracle Enterprise Session Border Controller.

In Survivable Mode, the Oracle Enterprise Session Border Controller routes incoming INVITEs based on the lookup from the registration cache. If the entry is part of the registration cache, the INVITEs are routed depending on the contact information from the cache. If the entry is not part of the registration cache, local policy is used if there is any local policy configured on the Oracle Enterprise Session Border Controller. The prefix length in the Survivability configuration is taken into consideration when creating the extension for the phone number in the registration cache.

Call Processing Behavior

After the Oracle Enterprise Session Border Controller enters Survivable Mode, it performs as follows for call processing:

- Allows incoming sessions (either from an endpoint or an external PSTN gateway or alternate trunk) to be processed locally, based on its Registration cache.
- Locally handles multiple identities based on the registered P-Preferred-Identity (or via BroadSoft's proprietary mechanism).
- For session requests coming from local endpoint destined to non-local destinations, it routes to alternate PSTN gateways or SIP trunks, if configured.
- It performs registration cache (reg-cache) matching based on substrings of the received dialed digits (for example, a phone registers as sip:7813284545@acmepacket.com and a local user dials sip:4545@acmepacket.com).

Note: The Oracle Enterprise Session Border Controller allows extensions to be dialed based on it's multiple user identities (identified either by using P-Asserted-Identity or BroadSoft's proprietary mechanism.) For more information about Survivability when using the BroadSoft server, see *Remote Site Survivability with a BroadSoft Server*

Exiting Survivable Mode

Registration Behavior

When the Remote Oracle Enterprise Session Border Controller exits Survivable Mode, it performs as follows for registrations:

- It forwards all registration requests (new or refreshes) to the core data center (or headquarters) site. Note: All endpoints in the registration cache associated with that Registrar are invalidated.
- "The "expires" value is no longer set to 30 seconds by default. It takes the corresponding registration-refresh value based on the Oracle Enterprise Session Border Controller configuration.

Note: When the Oracle Enterprise Session Border Controller is in Normal Mode, it routes the incoming INVITEs to the registrar if the endpoint is part of the registration cache. If the endpoint is not part of the registration cache, the INVITEs are routed using the local policy if the local policy is configured on the Oracle Enterprise Session Border Controller. Otherwise, a 404 Not Found is returned.

Call Processing Behavior

When the Remote Net-Net ESD exits Survivable Mode, it performs as follows for Call Processing:

- It allows incoming sessions to be sent to the core data center (or headquarters) site for processing.
- Existing sessions remain connected until a user ends the session.

Remote Site Survivability with a BroadSoft Server

The Remote Site Survivability feature can be enabled on a Oracle Enterprise Session Border Controller to work in a network with a BroadSoft server by installing the Survivability Session Plug-in Language (SPL) on the Oracle Enterprise Session Border Controller called BroadsoftSurvivability.spl.

In this network configuration, the Oracle Enterprise Session Border Controller advertises Directory Numbers (DNs), extensions, and other aliases (in XML format) in the 200 OK response to the Registrar. When the Oracle Enterprise Session Border Controller enters Survivability mode, an indication is sent to the BroadSoft server (as an XML object) in the 200 OK response in the REGISTER or SUBSCRIBE message. The Oracle Enterprise Session Border Controller then sends originations to all Shared Call Appearance (SCA) destinations via the BroadSoft server.

The following illustration shows the IP Phone sending a Register messsage through the Oracle Enterprise Session Border Controller to the BroadSoft server, and a 200 OK response returned from the BroadSoft server (containing the applicable XML info) through the Oracle Enterprise Session Border Controller to the IP Phone.

In the event that the BroadSoft server is unavailable, the Oracle Enterprise Session Border Controller creates a location mapping entry, linking the parsed information (DNs, extensions, and aliases) to the location cache entry's Address of Record (AOR). This allows users to dial by extension even if the BroadSoft server is unavailable.

Remote Site Survivability Configuration

You must enable remote site survivability on the Oracle Enterprise Session Border Controller (E-SBC) and set the ping method for the session agent before the E-SBC can perform remote site survivability operations.

The process for configuring remote site survivability includes the following procedures.

1. Enable remote site survivability mode on the E-SBC.
2. Configure a ping method for the session agent to use to determine when the E-SBC is not responding.

Note: The system does not require a reboot after activating or modifying remote site survivability.

Configuring a Service Tag for an IP Interface

To configure a service-tag for an IP interface:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. service-tag—Enter a character string that identifies a group of session-agents for the current SIP interface. When Survivability is enabled, the Oracle Enterprise Session Border Controller monitors the health of the session-agents using this service-tag.

```
ACMEPACKET(sip-interface)# service-tag intf1
```

5. Type done and press Enter.

```
ACMEPACKET(sip-interface)# done
ACMEPACKET(sip-interface)#
```

6. Type exit and press Enter.

```
ACMEPACKET(sip-interface)# exit
ACMEPACKET(session-router)#
```

7. Save the configuration.

Configure Remote Site Survivability

You must enable remote site survivability on the Oracle Enterprise Session Border Controller (E-SBC) and set the parameters before the system can enter and exit survival mode.

Prerequisites

- Confirm that at least one session agent is configured.

To enable remote site survivability from the ACLI command line, do the following :

1. From the ACLI command line, access the survivability object, and press Enter.

```
ACMEPACKET(session-router)# survivability
ACMEPACKET(survivability)#
```

2. state. Type enabled, and press ENTER.
3. reg-expires. Type a value for the number of seconds that the Oracle Enterprise Session Border Controller waits before entering the remote site survivability mode, and press ENTER.
4. prefix-length. Type the maximum number of digits allowed for a phone extension, and press ENTER. Valid values are 0-10.
5. session-agent hostname. Type the session agent hostname or the session agent group name, and press ENTER.
6. Type done, and press Enter.
7. Type exit, and press Enter.
8. Save the configuration.

Post-requisites

- Configure a ping method for the session agent to use to determine when the E-SBC is not responding.

Configuring Service Health for a List of Service Tag

To configure the service health for a list of service tags:

1. Type service-health and press Enter.

```
ACMEPACKET(session-router)# service-health
ACMEPACKET(service-health)#
```

2. Type service-tag and press Enter.

```
ACMEPACKET(service-health)# service-tag-list
ACMEPACKET(serviceTag)#
```

3. service-tag-string—Enter a list of service tags (associated with IP interfaces) on which the Oracle Enterprise Session Border Controller checks the service health. Default is blank.

```
ACMEPACKET(serviceTag)# service-tag-string intf1,intf2,intf3
```

4. Type sa-health-profile and press Enter.

```
ACMEPACKET(serviceTag)# sa-health-profile
ACMEPACKET(sa-health-profile)#
```

5. session-agent-hostname—Enter the hostname of the session agent on which the Oracle Enterprise Session Border Controller monitors the service health.

```
ACMEPACKET(sa-health-profile)# session-agent-hostname SA1
ACMEPACKET(sa-health-profile)#
```

6. session-agent-health—Enter the health score that the Oracle Enterprise Session Border Controller uses to determine whether or not the health of the session-agent has decremented and gone out of service or incremented and came back into service. Valid values are 0 to 100 percent. Default is 100. For example, if this parameter is set to 100, and if the health score of the session-agent falls beneath 100 percent, Survivability mode begins (if enabled). If the health of the session-agent comes back up to 100 percent, Survivability mode ends and the system returns to Normal mode.

Note: For cases where there are two session agents, each session agent could have a service health of 50.

```
ACMEPACKET(sa-health-profile)# session-agent-health 100
ACMEPACKET(sa-health-profile)#
```

7. Type done and press Enter.

```
ACMEPACKET(sa-service-health)# done
ACMEPACKET(sa-service-health)#
```

8. Type exit and press Enter.

```
ACMEPACKET(sa-service-health)# exit
ACMEPACKET(serviceTag)#
```

9. Type exit and press Enter.

```
ACMEPACKET(serviceTag)# exit
ACMEPACKET(service-health)#
```

10. Type exit and press Enter.

```
ACMEPACKET(service-health)# exit
ACMEPACKET(session-router)#
```

11. Save the configuration.

Configure the Ping Method for a Session Agent

Configure a ping method to confirm that the session agent is in service.

Use the session-agent object to configure the ping-method for a session-agent.

1. Access the session-agent object.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

2. ping-method—Type the SIP message/method to use to ping a session agent. Oracle recommends setting this value to OPTIONS.
3. ping-interval—Type the number of seconds between pings. The range is from 0-4294967295.
4. Type exit, and press Enter.
5. Save the configuration.

Example Remote Site Survivability Configuration

The following is an example of a Survivability mode configuration.

```
sip-interface
    service-tag        intf1
survivability
        state    enabled
        service-tag    intf1
        reg-expires    30
        prefix-length        4
service-health
    service-tag-list
        service-tag-string    intf1,intf2,intf3
        sa-health-profile
            session-agent-hostname    SA1
            session-agent-health    100
session-agent
    ping-method            BYE,ACK,OPTIONS,SUBSCRIBE,
                NOTIFY,INVITE,MESSAGE,INFO
```

Configuring Remote Site Survivability using the Web GUI

The Oracle Enterprise Session Border Controller Web GUI supports the configuration of Survivability.

Use the following procedure to configure Survivability.

Configure a Service Tag for an IP Interface

Configure a service tag to enable the Oracle Enterprise Session Border Controller to monitor the health of a group of session agents, when survivability is enabled.

- Confirm that survivability is enabled.
- Confirm that the system displays the Expert mode.

To configure a service-tag for an IP interface:

1. From the Web GUI, click **Configuration** > **session-router** > **sip-interface**.
2. On the Modify SIP Interface page, in the Service tag field, enter a character string that identifies a group of session-agents for the current SIP interface.
3. Click **OK**.
4. Save and activate the configuration.

Configure Remote Site Survivability

You must enable remote site survivability on the Oracle Enterprise Session Border Controller (E-SBC) and set the parameters before the system can enter and exit survival mode.

Before You Begin

- Confirm that at least one session agent is configured.
- Confirm that the system displays the Expert mode.

Procedure

1. From the Web GUI, click **Configuration** > **session-router** > **survivability**.
2. At the bottom of the left pane, click **Show advanced**.
3. On the Add survivability page, do the following:

Attributes	Instructions
State	Select to enable Survivability.
Reg expires	Enter the number of seconds that the Oracle Enterprise Session Border Controller waits before entering the remote site survivability mode when the registration expires.
Prefix length	Enter the maximum number of digits allowed for a phone extension. Range: 0-10.
Session agent hostname	Select the agent hostname or the session agent group name from the drop down list.

4. Click **OK**.
5. Save and activate the configuration.

Next Steps

- Configure a ping method on the session agent. See "Configure a Session Agent."

Configure Service Health

To configure the service health for a list of service tags:

1. Select **session-router** > **service-health.**
2. In the service-tag-list window, click <Add>.
3. In the service-tag-string field, enter a list of service tags (associated with IP interfaces) on which the Oracle Enterprise Session Border Controller (E-SBC) checks the service health. Default is blank. For example, intf1, intf2, intf3.
4. In the sa-health-profile box, click <Add>.
5. In the session-agent-hostname field, enter the hostname of the session agent on which the E-SBC monitors the service health.

6. In the session-agent-health field, enter the health score that the E-SBC uses to determine whether or not the health of the session-agent has decremented and gone out of service or incremented and came back into service. Valid values are 0 to 100 percent. Default is 100. For example, if this parameter is set to 100, and if the health score of the session-agent falls beneath 100 percent, Survivability mode begins (if enabled). If the health of the session-agent comes back up to 100 percent, Survivability mode ends and the system returns to Normal mode.

   Note: For cases where there are two session agents, each session agent could have a service health of 50.

7. Click <OK>.
8. Save and activate the configuration.

Configure the Ping Method for a Session Agent

Configure a ping method to confirm that the session agent is in service.

Use the session-agent object to configure the ping-method for a session-agent.

1. Click **Configuration** > **session-router** > **session-agent.**
2. On the Modify Session Agent page, select the session-agent for which you want to configure the ping-method, and click **OK**.
3. In the **Ping method** field, enter the SIP message/method to use to ping a session agent. Oracle recommends setting this value to OPTIONS.
4. In the **Ping interval** field, enter the number of seconds between pings.
5. Click**OK**.
6. Save and activate the configuration.

Show Commands for Survivability

The Oracle Enterprise Session Border Controller allows you to use specific show commands to display statisitcal data about Survivability mode. Survivability mode data consists of Session Initiation Protocol (SIP) Request method statistics. You can initiate the show commands whether or not the Oracle Enterprise Session Border Controller is in Survivability mode. However, if you initiate the commands when the Oracle Enterprise Session Border Controller is in Normal mode, and Survivability mode was never initiated, the statistics display as zero (0).

This section describes the various show ACLI commands you can use to display statistics about the performance of Survivability on the Oracle Enterprise Session Border Controller.

Show Survivability Command

The show survivability command displays active and total statistics about the performance of Survivability mode over a period of time and for overall lifetime. This display also provides statistics related to SIP media events that occur while the Oracle Enterprise Session Border Controller is in Survivability mode.

Note: The statistics that display in the output for this command are also used in the Historical Data Recording (HDR) statistics for Survivability. For more information about HDR for Survivability, see *Historical Data Recording (HDR) for Survivability*.

The following example shows the output for the show survivability command.

Example

```
ACMEPACKET# show survivability
12:44:48-109
SIP Status                 -- Period -- -------- Lifetime --------
                 Active    High   Total      Total  PerMax    High
Sessions              0       0       0          0       0       0
Subscriptions         0       0       0          0       0       0
Dialogs               0       0       0          0       0       0
CallID Map            0       0       0          0       0       0
Rejections            -       -       0          0       0
```

```
ReINVITEs              -        -        0          0        0
ReINV Suppress         -        -        0          0        0
Media Sessions         0        0        0          0        0        0
Media Pending          0        0        0          0        0        0
Client Trans           1        1        1        718        2        1
Server Trans           0        0        0          0        0        0
Resp Contexts          0        0        0          0        0        0
Saved Contexts         0        0        0          0        0        0
Sockets                2        2        0          2        2        2
Req Dropped            -        -        0          0        0
DNS Trans              0        0        0          0        0        0
DNS Sockets            0        0        0          0        0        0
DNS Results            0        0        0          0        0        0
Rejected Msgs          0        0        0          0        0        0
```

If Survivability mode was never initiated, the output shows values of zero (0) in all columns.

Output

The following table provides a description of this output.

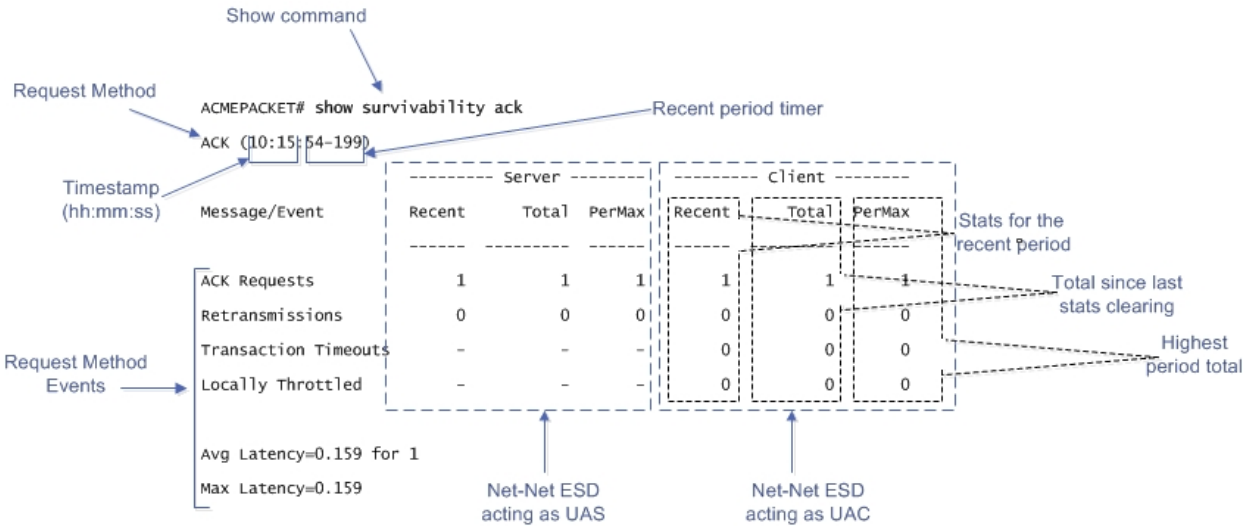
Event	Description
Sessions	Number of sessions established by INVITE and SUBSCRIBE messages during Survivability.
Subscriptions	Number of sessions established by SUBSCRIPTION during Survivability.
Dialogs	Number of end-to-end SIP signaling connections during Survivability.
CallID Map	Number of successful session header Call ID mappings during Survivability.
Rejections	Number of rejected INVITEs during Survivability.
ReINVITEs	Number of ReINVITEs during Survivability.
ReINV Suppress	Number of ReINVITEs that were suppressed during Survivability.
Media Sessions	Number of successful media sessions during Survivability.
Media Pending	Number of media sessions waiting to be established during Survivability.
Client Trans	Number of client transactions during Survivability.
Server Trans	Number of server transactions that have taken place on the Oracle Enterprise Session Border Controller during Survivability.
Resp Contexts	Number of response contexts during Survivability.
Saved Contexts	Number of saved contexts during Survivability.
Sockets	Number of SIP sockets during Survivability.
Req Dropped	Number of dropped requests during Survivability.
DNS Trans	Number of Domain Name System (DNS) transactions during Survivability.
DNS Sockets	Number of Domain Name System (DNS) sockets during Survivability.
DNS Results	Number of Domain Name System (DNS) results during Survivability.
Rejected Msgs	Number of rejected messages during Survivability.

Show Commands for Request Methods

The show survivability<method_name> command for SIP Request methods allow you to display specific statistical information about Request events that pass between the User Agent Server (UAS) and User Agent Client (UAC). Specific Request methods include:

SIP Request Method	Description
INVITE	Method used to request a session.
REGISTER	Method used to register the client with the server according to the address in the To header field.
BYE	Method used to terminate an established media session.
ACK	Method is used to acknowledge final responses to INVITE requests.
CANCEL	Method is used to terminate pending requests.
OPTIONS	Method used to query a user agent or server about its capabilities and discover its current availability.
REFER	Method used by a user agent to request another user agent to access a URI or URL resource.
SUBSCRIBE	Method used by a user agent to subscribe the device for the purpose of receiving notifications (via the NOTIFY method) about a particular event.
NOTIFY	Method used by a user agent to convey information about the occurrence of a particular event. A NOTIFY is always sent within a dialog, when a subscription exists between the subscriber and the notifier.
UPDATE	Method used to modify the state of a session without changing the state of the dialog,
PRACK	Method used to acknowledge receipt of reliably transported provisional responses. This is generated by a UAC.
MESSAGE	Method used to transport instant messages (IM) using SIP.
INFO	Method used to send information in the middle of a session that doesn't modify the session's state.
PUBLISH	Method used to publish an event state to the server.
OTHER	Method used

The following is an example of the show command output for an ACK Request.



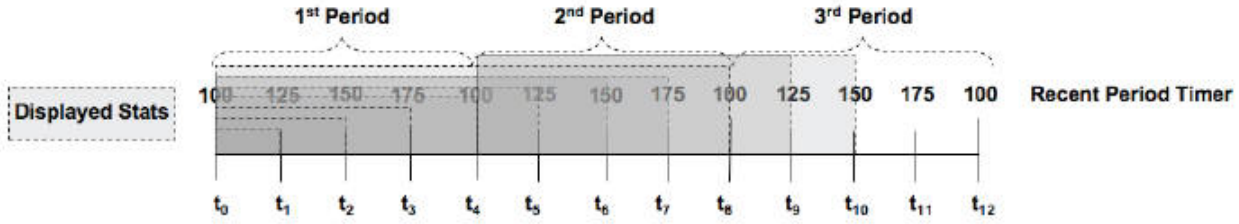
The example above provides a description for each area of the output. The Request method displays on the line directly under the command prompt (ACK in the above example), followed by the time stamp (hour:minute:second format), and then the recent period timer. The User Agent Server (UAS) data (when the Oracle Enterprise Session Border Controller is acting as a server) is listed in the middle of the display, and the User Agent Client (UAC) data (when the Oracle Enterprise Session Border Controller is acting as a client) is listed on the right side.

For both the UAS and UAC, the Recent column represents statistics for the recent period (the current period plus the last period). The Total column represents the total for a particular metric since the last stats clearing. Statistics are cleared either through the re-issue of the show survivability <method_name> command or on a reboot. The PerMax column represents the maximum for a given metric seen in any given individual (current) period.

Note: The "Recent" column represents the recent period, which includes statistics from the current and the last period, which is why that number may be higher than what displays in the PerMax column.

Recent Period Timer Operation

The Current period timer counts from 100 to 200 in one second increments as shown in the following illustration.



The statistics that display in the Recent column for any show survivability command reflects the appropriate behaviors for the associated value within the current period PLUS the last period (which constitutes a 100-200 second Recent period). This prevents the statistics from zeroing out between period transitions. So at time t4, in the display above, the statistics that display represent the last 100 seconds worth of behaviors (from the first period). The Recent Period statistics at time t6 represent the last 150 seconds of statistics (including 100 period 1). The Recent Period statistics at time t8 represent the last 100 seconds of statistics (including 100 from period 2).

The Recent period is the sum of the Active (current) period and the previous period.

SIP Request Method Examples

The following are examples of the show survivability <method_name> command. This command displays the recent and total Request events passed between the server and client when Survivability mode was enabled on the Oracle Enterprise Session Border Controller. This output also displays the maximum number of Request events that occurred during a current time period window of 100 seconds, when Survivability mode was enabled.

You can specify any SIP Request method for the <method_name>. The following example uses the INVITE SIP Request name.

Example 1

```
ACMEPACKET# show survivability invite
INVITE (10:15:44-189)
                        --------- Server --------   --------- Client --------
Message/Event           Recent      Total  PerMax   Recent      Total  PerMax
                        ------  ---------  ------   ------  ---------  ------
INVITE Requests              1          1       1        1          1       1
Retransmissions              0          0       0        0          0       0
100 Trying                   1          1       1        0          0       0
180 Ringing                  1          1       1        1          1       1
200 OK                       1          1       1        1          1       1
Response Retrans             0          0       0        0          0       0
Transaction Timeouts         -          -       -        0          0       0
Locally Throttled            -          -       -        0          0       0
Avg Latency=0.130 for 1
Max Latency=0.130
```

Example 2

The following example uses the REGISTER SIP Request name.

```
ACMEPACKET# show survivability register
REGISTER (09:55:26-150)
                        --------- Server --------   --------- Client --------
Message/Event           Recent      Total  PerMax   Recent      Total  PerMax
                        ------  ---------  ------   ------  ---------  ------
REGISTER Requests            4          4       4        4          4       4
Retransmissions              0          0       0        0          0       0
200 OK                       2          2       2        2          2       2
401 Unauthorized             2          2       2        2          2       2
Transaction Timeouts         -          -       -        0          0       0
Locally Throttled            -          -       -        0          0       0
Avg Latency=0.139 for 4
Max Latency=0.158
```

If Survivability mode was never initiated, the outputs show values of zero (0) in all columns.

show survivability commands

The following table describes the output for the “show survivability <method_name> command.

Message/Event	Description
INVITE Requests	Number of INVITE Request events that occurred between the server and client during Survivability mode.
Retransmissions	Number of retransmission of INVITE Request events that occurred during Survivability.
<Response Code>	Type and number of responses that occurred between the Client and Server during Surviability.
Transaction Timeouts	Number of INVITE Request event timeouts that occurred during Survivability.
Locally Throttled	Number of INVITE Request events that were locally throttled during Survivability. This is the number of INVITE Request events that were transmitted during the regulation (slowing down) of network traffic by the Oracle Enterprise Session Border Controller to minimize bandwidth congestion.

Message/Event	Description
Avg Latency	Average amount of time for INVITE Request events to travel in the time period window with the amount of events specified.
Max Latency	Maximum amount of time it took for INVITE Request events to travel in the time period window.

Show Commands for Session Agents Interfaces and Realms

The following show commands for Session Agents, interfaces and realms allow you to display recent and total statistics about the SIP methods used during Survivability mode:

- show survivability agents <hostname><method_name>
- show survivability interface <realm-id><method_name>
- show survivability realms <realm-id><method_name>

For each of these commands you can specify the SIP method name for which you want to display statistics. SIP method names include:

BYE	OPTIONS
UPDATE	SUBSCRIBE
CANCEL	NOTIFY
ACK	INFO
INVITE	MESSAGE
PRACK	PUBLISH
REFER	REGISTER
OTHER	

The output for these commands display recent and total number of SIP Requests that occurred for a session agent, interface, or realm during a current time period window of 100 seconds, when Survivability mode was enabled.

Note: To view the method names available, press the tab key after entering the command as shown in the following example.

```
ACMEPACKET# show survivability agents net192<tab>
ack          bye          cancel       info         invite       message
notify       options      other        prack        publish      refer
register     subscribe    update
```

The following examples show the output of the show survivability commands for agents, interface, and realms.

If Survivability mode was never initiated, the outputs show values of zero (0) in all columns.

Session Agents

```
ACMEPACKET# show survivability agents net192 refer
REFER (13:15:35-117)
                    --------- Server --------   --------- Client --------
Message/Event       Recent      Total   PerMax   Recent      Total   PerMax
                    ------  ---------  ------   ------  ---------  ------
REFER Requests           0          2       2        0          2       2
Retransmissions          0          0       0        0          0       0
202 Accepted             0          2       2        0          2       2
Transaction Timeouts     -          -       -        0          0       0
Locally Throttled        -          -       -        0          0       0
```

```
Avg Latency=0.000 for 0
Max Latency=0.000
```

Interface

```
ACMEPACKET# show survivability interface net192 refer
REFER (13:15:35-117)
                     --------- Server --------   --------- Client --------
Message/Event        Recent      Total  PerMax   Recent      Total  PerMax
                     ------  ---------  ------   ------  ---------  ------
REFER Requests            0          2       2        0          2       2
Retransmissions           0          0       0        0          0       0
202 Accepted              0          2       2        0          2       2
Transaction Timeouts      -          -       -        0          0       0
Locally Throttled         -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
```

Realms

```
ACMEPACKET# show survivability realms net192 refer
REFER (13:15:35-117)
                     --------- Server --------   --------- Client --------
Message/Event        Recent      Total  PerMax   Recent      Total  PerMax
                     ------  ---------  ------   ------  ---------  ------
REFER Requests            0          2       2        0          2       2
Retransmissions           0          0       0        0          0       0
202 Accepted              0          2       2        0          2       2
Transaction Timeouts      -          -       -        0          0       0
Locally Throttled         -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
```

Output

The following table describes the output for the above commands.

Message/Event	Description
<method_name> Requests	Number of the specified Request events that occurred between the server and client during Survivability mode.
Retransmissions	Number of retransmissions of specified Request message that occurred during Survivability.
<Response Code>	Type and number of responses that occurred between the Client and Server during Surviability.
Transaction Timeouts	Number of the specified Request event timeouts that occurred during Survivability.
Locally Throttled	Number of the specified Request events that were locally throttled during Survivability. This is the number of ACK Request events that were transmitted during the regulation (slowing down) of network traffic by the Oracle Enterprise Session Border Controller to minimize bandwidth congestion.
Avg Latency	Average amount of time for the specified Request events to travel in the time period window of 100 seconds, for the amount of events specified, during Survivability.
Max Latency	Maximum amount of time it took for the specified Request events to travel in the time period window of 100 seconds during Survivability.

Show Command for Survivability Status

The show survivability status command allows you to display the current status of Survivability mode on the Oracle Enterprise Session Border Controller. This command displays whether or not Survivability mode is enabled on an interface, and the date and time that Survivability mode was enabled.

The following is an example output of the show survivability status command.

Example

```
ACMEPACKET# show survivability status
Survivability
sip-interface  service-tag  state        start time       end time
-------------------------------------------------------------------------
net192         test         enabled      Aug 15 12:53:01  -
net172         none         n/a          n/a              n/a
```

The following table describes the output for the above command.

Column	Description
sip-interface	Interface currently configured on the Net-Net ESD.
service-tag	Service tag that indicates the Session Agent Group (SAG) assigned to the interface on the Oracle Enterprise Session Border Controller.
state	Current Survivability state on the interface. Valid values are: enabled - Survivability is enabled on the interface disabled - Survivability is disabled on the interface n/a - Survivability does not configured on this interface.
start time	The date (MM:DD) and time (HH:MM:SS) that Survivability Mode became in-service on the interface.
end time	The date (MM:DD) and time (HH:MM:SS) that Survivability Mode became out-of-service on the interface. A - indicates that Survivability Mode is currently in-service and has not yet ended.

You can also display the current status of Survivability mode on a specific interface using the command, show survivability status <interface> where <interface> is the SIP interface name.

The following is an example output of the show survivability status <interface> cocmmand.

```
ACMEPACKET# show survivability status net192
Survivability
sip-interface  service-tag  state         start time       end time
-------------------------------------------------------------------------
net192         test         enabled       Aug 15 12:53:01  -
```

Show Command for Service Health

When Survivability Mode is active on the Oracle Enterprise Session Border Controller, the system is able to detect any loss of connection (and subsequent re-connection) to the Enterprise's core data center based on a health score. The health score is the value that the Oracle Enterprise Session Border Controller uses to determine whether or not the health of the session-agent has decremented and gone out of service or incremented and came back into service.

You configure the service health at the ACLI path sessions-router->service-health ->service-tag-list->sa-health-profile->session-agent-health. Valid values are 0 to 100 percent. Default is 100. For example, if the session-agent-health parameter is set to 100, and if the health score of the session-agent falls beneath 100 percent, Survivability mode becomes in-service (if enabled). If the health of the session-agent comes back up to 100 percent, Survivability mode goes out of service, and the system returns to Normal mode.

For more information about service health in relation to Survivability on the Oracle Enterprise Session Border Controller, see *Survivability Health Score* .

You can display the current service-health of Survivability mode using the command show service-health at the root prompt.

If service-health on the Oracle Enterprise Session Border Controller is configured as follows:

```
service-health
        service-tag
                service-tag-string                      test
                sa-health-profile
                        session-agent-hostname      testAgent
                        session-agent-health     100
        last-modified-by                        admin@console
        last-modified-date                      2013-07-23 10:31:48
```

then the following are example outputs of the show service-health cocmmand when Survivability mode is in-service and out-of-service on the Oracle Enterprise Session Border Controller.

In-Service Example

```
ACMEPACKET# show service-health
service-tag      healthScore
test     100
```

Out-of-Service Example

```
ACMEPACKET# show service-health
service-tag      healthScore
test     0
```

Historical Data Recording (HDR) for Survivability

If the Oracle Enterprise Session Border Controller is configured to collect Historical Data Recording (HDR) statistics, statistics are collected on Survivability whether or not it is in-service.

HDR data consists of a “Group” with associated Group Statistics that apply to each group. HDR data comes from two sources:

- Simple Network Management Protocol (SNMP) Management Information Bases (MIBs)
- Oracle’s Command Line Interface (ACLI)

The Survivability data in the HDR outputs are taken from the ACLI. The following are the HDR Groups for survivability:

- survivability-sip-status
- survivability-sip-invites
- survivability-sip-register
- survivability-sip-errors

When the collector on the Oracle Enterprise Session Border Controller is enabled, these Groups and associated Group Statistics are included in the collection of data.

The following paragraphs provide a description of each Survivability Group and Group Statistic. Each Group table identifies the ACLI Show command for which it is associated.

Group survivability-sip-status

Description	Consists of statistics pertaining to the status of Survivability on the Oracle Enterprise Session Border Controller.
Group Statistics	*Sessions* *Subscriptions* *Dialogs* *CallID Maps* *Rejections* *ReINVITEs* *Media Sessions* *Media Pending* *Client Trans* *Server Trans* *Resp Contexts* *Saved Contexts* *Sockets* *Req Drops* *DNS Trans* *DNS Sockets* *DNS Results* *Session Rate* *Load Rate* *Active Subscriptions* *SubscriptionsPerMax* *Subscriptions High*
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Group Statistics

Active Subscriptions

Description	Specifies the current global count of active SIP subscriptions during Survivability.
Type	counter

Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command show sipd realms <realm_name> in the Net-Net SBC Historical Data Recording Resource Guide.

CallID Maps

Description	Total number of successful session header Call ID mappings during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Client Trans

Description	Total number of client transactions during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

DNS Results

Description	Total number of Domain Name System (DNS) results during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability

ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

DNS Sockets

Description	Total number of Domain Name System (DNS) sockets during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

DNS Trans

Description	Total number of Domain Name System (DNS) transactions during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Dialogs

Description	Total number of end-to-end SIP signaling connections during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Load Rate

Description	Average Central Processing Unit (CPU) utilization of the Oracle Enterprise Session Border Controller during the current window period, and during Survivability. The average is computed every 10 seconds unless the load-limit is configured in the SIPConfig record, in which case it is 5 seconds.
Type	period
Timer Value (seconds)	30
Range	0% to 100%
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Media Pending

Description	Total number of media sessions waiting to be established during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Media Sessions

Description	Total number of successful media sessions during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

ReINVITEs

Description	Total number of ReINVITEs during Survivability.
Type	counter

Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Rejections

Description	Total number of rejected INVITEs during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Req Drops

Description	Total number of dropped requests during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Resp Contexts

Description	Total number of response contexts during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability

ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Saved Contexts

Description	Total number of saved contexts during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Server Trans

Description	Total number of server transactions that have taken place on the Oracle Enterprise Session Border Controller during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Sessions

Description	Total number of sessions established by INVITE and SUBSCRIBE messages during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Session Rate

Description	The rate, per second, of SIP invites allowed to or from the Oracle Enterprise Session Border Controller during the sliding window period, and during Survivability. The rate is computed every 10 seconds .
Type	period
Timer Value (seconds)	30
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Sockets

Description	Total number of SIP sockets during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Subscriptions

Description	Total number of sessions established by SUBSCRIPTION during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability
ACLI Parameter Mapping	For ACLI parameter mappings, see the table at *Show Survivability Command* .

Subscriptions High

Description	Specifies the maximum global count of active SIP subscriptions since the last SBC re-boot, and during Survivability.
Type	counter

Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command show sipd realms <realm_name> in the Net-Net SBC Historical Data Recording Resource Guide.

SubscriptionsPerMax

Description	Specifies the maximum global count of SIP subscriptions initiated during any 100 second period since the last SBC re-boot, and during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command show sipd realms <realm_name> in the Net-Net SBC Historical Data Recording Resource Guide.

Group survivability-sip-invites

Description	**Consists of response statistics pertaining to INVITES during Survivability on the Oracle Enterprise Session Border Controller.**
Group Statistics	***INVITE Requests*** ***Retransmissions*** ***Response Codes*** **Each response code is next printed to the HDR file on a separate line. The format is <timestamp> <3-digit-code Description> <Total count> <Client total count>. See the above link to the Response Codes description table.** ***Response Retrans*** ***Transaction Timeouts*** ***Locally Throttled***
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Group Statistics

INVITE Requests

Description	Total number of INVITE requests during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Locally Throttled

Description	Total number of INVITE requests locally throttled during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Response Codes

Description	Total number of a specific INVITE response codes that occurred during Survivability. Each of the response codes are as follows: 1xx --Informational: 100 Trying: This response is used to indicate the next node receives the request and stop the retransmission. This response is sent if there is delay in sending the final response more the 200ms. 180 Ringing: The response is generated if UA receives the INVITE and started the ringing. It may used to initiate local ring back. 181 Call is being Forwarded: This response is indication of call is being forwarded to different destination. 182 Call Queued: The called server is overloaded or temporary unavailable. the server sends this status code to queue the call. When server ready to take the call, it initiates appropriate final response. 183 Call Progress: This response may be used to send extra information for a call which is still being set up.

	2xx—Successful Responses 200 OK: Indicates the request was successful. 202 Accepted: Indicates that the request has been accepted for processing, but the processing has not been completed. 3xx—Redirection Response 301 Moved Permanently: The original Request-URI is no longer valid, the new address is given in the Contact header field, and the client should update any records of the original Request-URI with the new value. 302 Moved Temporarily: The client should try at the address in the Contact field. If an Expires field is present, the client may cache the result for that period of time. 305 Use Proxy: The Contact field details a proxy that must be used to access the requested destination. 380 Alternative Service: The call failed, but alternatives are detailed in the message body. 4xx—Client Failure Responses 400 Bad Request: The request could not be understood due to malformed syntax. 401 Unauthorized: The request requires user authentication. This response is issued by UASs and registrars. 403 Forbidden: The server understood the request, but is refusing to fulfill it 404 Not Found: The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request. 405 Method Not Allowed: The method specified in the Request-Line is understood, but not allowed for the address identified by the Request-URI. 406 Not Acceptable: The resource identified by the request is only capable of generating response entities that have content characteristics but not acceptable according to the Accept header field sent in the request. 407 Proxy Authentication Required: The request requires user authentication. This response is issued by proxys
	4xx—Client Failure Responses (continued) 408 Request Timed Out: Couldn't find the user in time. 415 Unsupported Media Type: Request body in a format not supported. 420 Bad Extension: Bad SIP Protocol Extension used, not understood by the server. 421 Extension Required: The server needs a specific extension not listed in the Supported header. 422 Session Interval Too Small: The received request contains a Session-Expires header field with a duration below the minimum timer. 423 Interval Too Brief: Expiration time of the resource is too short. 480 Temporarily Unavailable: Callee currently unavailable. 481 Call/Transaction Does Not Exist: Server received a request that does not match any dialog or transaction. 482 Loop Detected: Server has detected a loop. 483 Too Many Hops: Max-Forwards header has reached the value '0'.

	484 Address Incomplete: Request-URI incomplete. 485 Ambiguous: Request-URI is ambiguous. 486 Busy Here: Callee is busy. 487 Request Terminated: Request has terminated by bye or cancel. 488 Not Acceptable Here: Some aspects of the session description of the Request-URI is not acceptable. 489 Bad Event: The server did not understand an event package specified in an Event header field. 491 Request Pending: Server has some pending request from the same dialog.
	5xx—Server Failure Responses 500 Server Internal Error: The server could not fulfill the request due to some unexpected condition. 501 Not Implemented: The server does not have the ability to fulfill the request, such as because it does not recognize the request method. (Compare with 405 Method Not Allowed, where the server recognizes the method but does not allow or support it.) 502 Bad Gateway: The server is acting as a gateway or proxy, and received an invalid response from a downstream server while attempting to fulfill the request. 503 Service Unavailable: The server is undergoing maintenance or is temporarily overloaded and so cannot process the request. A "Retry-After" header field may specify when the client may re attempt its request. 504 Server Time-out: The server attempted to access another server in attempting to process the request, and did not receive a prompt response. 513 Message Too Large: The request message length is longer than the server can process. 580 Precondition Failure: The server is unable or unwilling to meet some constraints specified in the offer. 6xx—Global Failure Responses 600 Busy Everywhere: All possible destinations are busy. Unlike the 486 response, this response indicates the destination knows there are no alternative destinations (such as a voicemail server) able to accept the call. 603 Decline: The destination does not wish to participate in the call, or cannot do so, and additionally the client knows there are no alternative destinations (such as a voicemail server) willing to accept the call. 604 Does Not Exist Anywhere: The server has authoritative information that the requested user does not exist anywhere. 606 Not Acceptable: The user's agent was contacted successfully but some aspects of the session description such as the requested media, bandwidth, or addressing style were not acceptable.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite

ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Response Retrans

Description	Total number of INVITE response retransmissions during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Retransmissions

Description	Total number of retransmissions of INVITEs during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Transaction Timeouts

Description	Total number of INVITE request transaction timeouts during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Group survivability-sip-register

Description	Consists of response statistics pertaining to REGISTRATIONS during Survivability on the Oracle Enterprise Session Border Controller.
Group Statistics	***REGISTRATION Requests*** ***Retransmissions*** ***Response Retrans*** ***Transaction Timeouts*** ***Locally Throttled***
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability register” at *SIP Request Method Examples* .

Group Statistics

Locally Throttled

Description	Total number of Register requests locally throttled during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Group: survivability-sip-errors

Description	Consists of response statistics pertaining to REGISTRATIONS during Survivability on the Oracle Enterprise Session Border Controller.
Group Statistics	*SDP Offer Errors* *SDP Answer Errors* *Drop Media Errors* *Transaction Errors* *Application Errors* *Media Exp Events* *Early Media Exps* *Exp Media Drops* *Expired Sessions* *Multiple OK Drops* *Multiple OK Terms* *Media Failure Drops* *Non-ACK 2xx Drops* *Invalid Requests* *Invalid Responses* *Invalid Messages* *CAC Session Drop* *CAC BW Drop*
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

REGISTRATION Requests

Description	Total number of Register requests sent between the client and server during Surivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability register” at *SIP Request Method Examples* .

Response Retrans

Description	Total number of Register response retransmissions during Survivability.

Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Retransmissions

Description	Total number of Register retransmissions that ocurred during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability register
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability register” at *SIP Request Method Examples* .

Transaction Timeouts

Description	Total number of Register request transaction timeouts during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability invite
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability invite” at *SIP Request Method Examples* .

Group Statistics

Application Errors

Description	Total number of miscellaneous errors in the SIP application that are otherwise uncategorized during Survivability.
Type	counter

Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples*.

CAC BW Drop

Description	Total number of call admission control (CAC) session setup failures due to insufficient bandwidth (BW) during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

CAC Session Drop

Description	Total number of call admission control (CAC) session setup failures during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Drop Media Errors

Description	Total number of errors encountered during Survivability, in tearing down the media for a dialog or session that is being terminated due to: a) non-successful response to an INVITE transaction, or b) a BYE transaction received from one of the participants in a dialog/session, or c) a BYE initiated by the Net-Net SD due to a timeout notification from the Middlebox Control Daemon (MBCD).
Type	counter

Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Early Media Exps

Description	Total number of flow timer expiration notifications received for media sessions that were not completely set up due to an incomplete or pending INVITE transaction during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Expired Sessions

Description	Total number of sessions terminated due to the session timer expiring during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Exp Media Drops

Description	Total number of flow timer expiration notifications from the Middlebox Control Daemon (MBCD) that resulted in the termination of the dialog/session by the SIP application during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295

ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Invalid Messages

Description	Total number of messages dropped due to parse failure during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Invalid Requests

Description	Total number of invalid requests (for example, an unsupported header was received) during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Invalid Responses

Description	Total number of invalid responses (for example, no Via header in response) during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Media Exp Events

Description	Total number of flow timer expiration notifications received from the Middlebox Control Daemon (MBCD) during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Media Failure Drops

Description	Total number of dialogs terminated due to a failure in establishing the media session during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Multiple OK Drops

Description	Total number of dialogs terminated upon reception of a 200 OK response from multiple User Agent Servers (UASs) for a given INVITE transaction that was forked by a downstream proxy during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Multiple OK Terms

Description	Total number of dialogs terminated upon reception of a 200 OK response that conflicts with an existing established dialog on the Oracle Enterprise Session Border Controller during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

Non-ACK 2xx Drops

Description	Total number of sessions terminated because an ACK was not received for a 2xx response during Survivability.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

SDP Answer Errors

Description	Total number of errors encountered during Survivability, in setting up the media session for a session description in a SIP request or response which is a Session Description Protocol (SDP) Answer in the Offer/Answer model (RFC 3264)
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

SDP Offer Errors

Description	Total number of errors encountered during Survivability, in setting up the media session for a session description in a SIP request or response which is a Session Description Protocol (SDP) Offer in the Offer/Answer model (RFC 3264).
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

SNMP Trap for Survivability

A Oracle Enterprise Session Border Controller MIB contains objects of management data, and also information about Simple Network Management Protocol (SNMP) traps, which enable an agent to notify the management station of significant events by way of an unsolicited SNMP message. When an element sends a TRAP packet, it can include an Object Identifier (OID) and value information (bindings) to clarify the event. For more information about SNMP on the Oracle Enterprise Session Border Controller, see the Net-Net 4000 MIB Reference Guide.

The Oracle Enterprise Session Border Controller triggers an Enterprise SNMP trap when a SIP interface goes in or out of Survivability mode. This trap is called:

- snmp_survivability_mode_trap_send

This trap has been added to the SIP application MIB called ap-sip.mib. The trap information is as follows in this MIB:

```
apSipSurvivabilityNotif                OBJECT IDENTIFIER ::=
{ apSipNotificationObjects 2 }
apSipSurvivabilityNotifObjects      OBJECT IDENTIFIER ::=
{ apSipSurvivabilityNotif 1 }
apSipSurvivabilityNotifPrefix       OBJECT IDENTIFIER ::=
{ apSipSurvivabilityNotif 2 }
apSipSurvivabilityNotifications     OBJECT IDENTIFIER ::=
{ apSipSurvivabilityNotifPrefix 0 }
apSipSurvivabilityModeEnter     NOTIFICATION-TYPE
    OBJECTS    { apSysMgmtSipInterfaceRealmName,
apSysMgmtSipInterfaceIP  }
    STATUS        current
    DESCRIPTION
    " The trap will be generated when SIP interface enters Survivability
Mode."
    ::= { apSipSurvivabilityNotifications 1 }
apSipSurvivabilityModeExit         NOTIFICATION-TYPE
    OBJECTS    { apSysMgmtSipInterfaceRealmName,
         apSysMgmtSipInterfaceIP  }
    STATUS        current
    DESCRIPTION
        " The trap will be generated when SIP interface exits Survivability
Mode and resumes normal operation."
    ::= { apSipSurvivabilityNotifications 2 }
apSipSurvivabilityNotificationsGroup     NOTIFICATION-GROUP
        NOTIFICATIONS {apSipSurvivabilityModeEnter,
            apSipSurvivabilityModeExit  }
        STATUS      current
        DESCRIPTION
```

```
        "Traps to monitor  SIP interface Survivability feature."
        ::= { apSipNotificationGroups  2 }
```

Note: The apSysMgmtSipInterfaceRealmName and apSysMgmtSipInterfaceIP objects are imported strings defined in ap-smgmt.mib.

When this trap is generated, it contains the following information:

realmname	Realm name of the SIP interface
ipaddr	IP address of the SIP interface
mode	Specifies whether or not the SIP interface is in survivability mode. Values included with the trap are: 0 - SIP interface is OK. It is not in Survivability mode. 1 - SIP interface is in Survivability mode.

The following table identifies the Survivability OBJECT IDENTIFIERS in the Oracle MIB that the Oracle Enterprise Session Border Controller supports.

Trap Name: OID Number	Description
apSipSurvivabilityNotificationsGroupCap: 1.3.6.1.4.1.9148.2.1.21.3	Specifies the capability of the Oracle Enterprise Session Border Controller to notify the Agent regarding Survivability on the SIP interface.
apSipSurvivabilityNotif 1.3.6.1.4.1.9148.3.15.2.2	N/A
apSipSurvivabilityNotifObjects 1.3.6.1.4.1.9148.3.15.2.2.1	N/A
apSipSurvivabilityNotifPrefix 1.3.6.1.4.1.9148.3.15.2.2.2	N/A
apSipSurvivabilityNotifications 1.3.6.1.4.1.9148.3.15.2.2.2.0	N/A
apSipSurvivabilityModeEnter 1.3.6.1.4.1.9148.3.15.2.2.2.0.1	Specifies that the SIP interface has entered Survivability mode.
apSipSurvivabilityModeExit 1.3.6.1.4.1.9148.3.15.2.2.2.0.2	Specifies that the SIP interface has exited Survivability mode and resumed normal operation.
apSipSurvivabilityNotificationsGroup 1.3.6.1.4.1.9148.3.15.3.2.2	Specifies the notification from the Oracle Enterprise Session Border Controller to the Agent regarding Survivability on the SIP interface.

Survivability Alarms and Logging

All survivability debug information and messages are logged to the serviceHealth.log. When a SIP interface enters Survivability Mode, a MAJOR alarm is raised. The alarm message contains the SIP interface's IP address and realm ID on which it resides. The following is an example of the alarm.

Survivability Alarm Example

```
ID     Task     Severity     First Occurred     Last Occurred
3145745     776175088     4          2013-08-20 10:19:35      2013-08-20 10:19:35
Count     Description
1     SIP interface ip=172.16.38.17 realm-id=core running in Survivability Mode
```

Transaction Errors

Description	Total number of errors encountered during Survivability when processing SIP client transactions associated with setting up or tearing down of the media session.
Type	counter
Timer Value (seconds)	N/A
Range	0 to 4294967295
ACLI Show Command	show survivability errors
ACLI Parameter Mapping	For ACLI parameter mappings, see the command “show survivability errors” at *SIP Request Method Examples* .

25

Emergency Location Identification Number (ELIN) Gateway Support

An ELIN-capable gateway supports connection to a qualified E911 service provider. The connection supports PSTN-based E911 functions, including user callback when there is a disconnect. Enterprises often deploy ELIN numbers based on physical location to locate the physical source of a 911 call. By using multiple ELINs, an enterprise can support multiple, simultaneous E911 calls.

Typically, an enterprise purchases multiple ELIN numbers. An ELIN gateway replaces VoIP extension URIs with ELIN numbers and maintains the mapping. For example, if an emergency service replied to a VoIP URI without using an ELIN gateway, the reply would be delayed or fail. An ELIN gateway can use its mapping to translate the ELIN number back to the VoIP extension from within the enterprise session network. The gateway can immediately forward the call back to the original client.

The Oracle Enterprise Session Border Controller supports E911 ELIN for Lync-enabled Enterprises using the ELIN_Gateway SPL option. Enable this option in the global SPL configuration. The Oracle Enterprise Session Border Controller supports up to 300 ELIN numbers simultaneously and it can reuse numbers allowing a greater number of emergency calls.

How the Emergency Location Identification Number (ELIN) SPL Works

When a Lync client places a 911 emergency call through a mediation server to a Oracle Enterprise Session Border Controller, the server indicates the emergency status in the priority field and provides a list of ELIN numbers. When the ELIN gateway module is enabled, the Oracle Enterprise Session Border Controller intelligently selects a particular ELIN number and uses it as the ANI in the “From” field SIP URI in the outgoing INVITE.

The Oracle Enterprise Session Border Controller preserves the mapping of used ELIN numbers in an internal table. This table includes the ELIN number, the caller (VoIP extension), the “in-use” count, and a timer field. The Oracle Enterprise Session Border Controller retains these mappings for a configurable time period ranging from 30 to 60 minutes after the call is terminated. The default is 30 minutes. When the timer expires, the entry is purged from the table. The timer field shows the time of day that the timer started.

You can view the current ELIN table at any time using the ACLI command spl show sip elins.

After the Lync client call is disconnected, the 911 service may call back using the number provided in the “From” field of the original INVITE. This presence of this number in its ELIN number table allows the Oracle Enterprise Session Border Controller to route the call back to the original caller.

Number Reuse

The Oracle Enterprise Session Border Controller can use an ELIN number for multiple calls. When a call that requires an ELIN mapping arrives at the Oracle Enterprise Session Border Controller, it checks to see if the numbers presented by the mediation server are in use. If a number is not in use, it simply uses that number. A number is not in use if it is not in the table or its "used count" is 0. An entry's used count is zero when its not in use and its purge timer has not yet expired.

If all numbers are in use, the Oracle Enterprise Session Border Controller employs a means of reusing a number, incrementing its used count for each additional call. The selection process proceeds in the following order:

1. If the "caller" is in the ELIN table, the Oracle Enterprise Session Border Controller selects that mapping.
2. The Oracle Enterprise Session Border Controller selects the number with the lowest "ELIN count".

If an ELIN number is used by multiple calls, it maps callback attempts to that ELIN number to the client that was last associated with the number.

Error Handling

Lync mediation servers always expect 503 "Service Unavailable" as an error message to a failed ELIN call. There is a variety of error messages that the network may send back when a call fails. For the purposes of Lync support, the Net-Net ESD sends 503 "Service Unavailable" to indicate call failure to a mediation server, regardless of the error it receives.

Configure the Emergency Location Identification Number (ELIN) Gateway Option

The ELIN-Gateway option must be configured at the global level under spl-config or by way of the Web GUI. The ELIN-Gateway option is not recognized in the session-agent, realm-config, or sip-interface.

Determine the preferred length of time to retain ELIN mappings within the Oracle Enterprise Session Border Controller. The range is from 30 to 60 minutes. The default is 30 minutes.

To configure the ELIN Gateway option:

1. In Superuser mode, type configure terminal and press <Enter>.
   ACMEPACKET# configure terminal
2. Type system, and press <Enter>.

   ```
   ACMEPACKET(configure)# system
   ACMEPACKET(system)#
   ```

3. Type spl-config, and press <Enter>.

   ```
   ACMEPACKET(configure)# spl-config
   ACMEPACKET(spl-config)#
   ```

4. Type spl-options +Extension-Headers="<value>" where <value> is the additional header information to store, and press <Enter>. The default behavior stores only the Request-URI and realm-id.

   ```
   ACMEPACKET(spl-config)#spl-options +Elin-Gateway=60
   ```

5. Type done to save your work.

The following is an example of an Elin_Gateway SPL configuration:

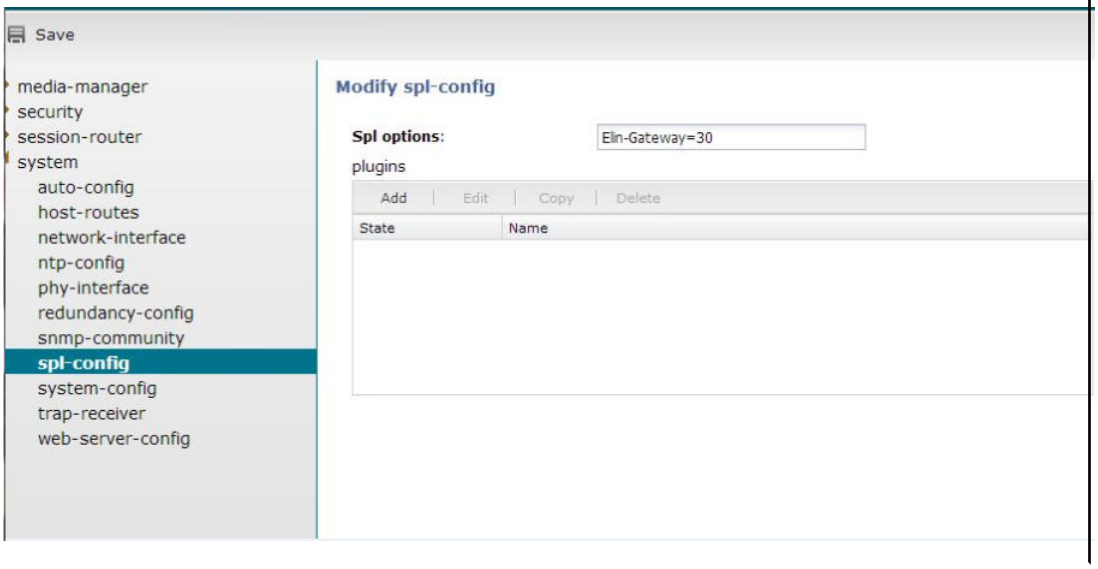
```
system
    spl-config
        spl-options          Elin-Gateway=60
```

The following is an example of the ACLI command spl show sip elins.

```
ACMEPACKET#show sip elins
Elin:1111442231
Count:0 From:5555221134 Time:1380490337.8292
```

```
--------------
Elin:2222882232
Count:0 From:6666111234 Time:1380490770.4083
```

To configure the ELIN-Gateway option using the Web GUI, select spl-config, add a config, and save.



26

Avaya Session Manager (SM) Redundancy

To support redundancy in Avaya SM deployments, the Oracle Enterprise Session Border Controller can use the mechanisms for maintaining multiple connections defined in RFC 5626. In an Avaya SM deployment, this scenario is referred to as Dual Registration.

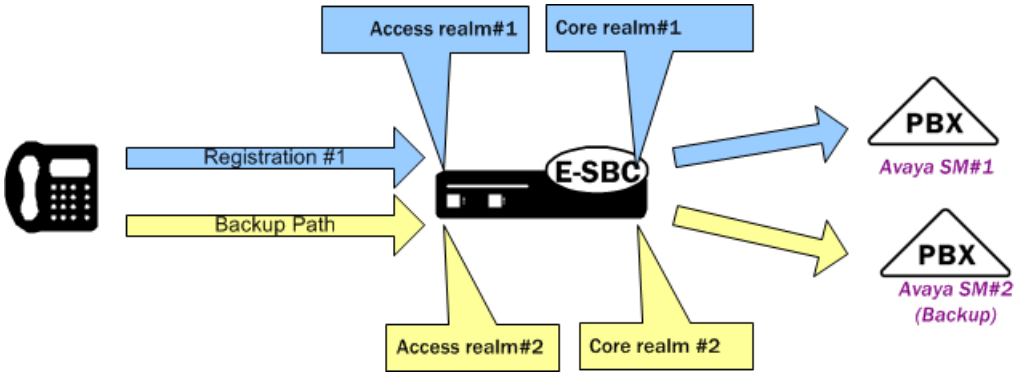
RFC 5626 specifies a method of maintaining connections between UAs and proxies, and outlines a general means for UAs to establish connection redundancy. The Oracle Enterprise Session Border Controller can use RFC 5626 specifically for redundancy in Avaya SM Dual Registration deployments. Such a deployment allows the network to continue to provide service by way of a redundant Avaya SM, when the primary Avaya SM stops responding.

Oracle Enterprise Session Border Controller configuration requires adding the rfc 5626 SPL option. In addition to adding the SPL option, the Oracle Enterprise Session Border Controller configuration design separates Avaya SMs and UA traffic by way of using realms.

How Avaya Session Manager (SM) Redundancy Works

To support Avaya SM redundancy, you configure multiple realms on the access side and the core side of the Oracle Enterprise Session Border Controller. These realms create the primary path and backup path for accessing a redundant Avaya SM.

Consider two Avaya SMs deployed for redundancy. You configure a core side realm for each Avaya SM and you configure two access side realms. Each access side realm is associated with the applicable core side realm, to which a UA sends registration messaging. The following illustration shows this configuration.



The operational scenario consists of the Avaya SM infrastructure providing configuration information to the UAs. The information includes the 2 proxy addresses, targeting the Oracle Enterprise Session Border Controller access-side interfaces. The UA knows which proxy is for the primary path, and sends initial REGISTER messages by way of that

path. While the primary Avaya SM is up, the UA manages all registration exchanges, including refresh and re-register procedures, on the primary path. If the primary Avaya SM stops responding, the infrastructure informs the UA that it needs to register with the backup Avaya SM. The UA registers with the backup Avaya SM using the backup path.

The UA, by way of configuration, populates the backup registration and subsequent registration messages so that the Avaya SM infrastructure knows that the registrations are for the same UA. Key elements of the messaging and their use by the Avaya dual registration infrastructure include:

- reg-id - A contact header field parameter value that specifies individual registrations. UAs use unique reg-id values to specify registrations for individual flows.
- instance-id (+sip.instance) - A URN within the contact header that specifies a UA instance. UAs use the same Instance ID information in REGISTER exchanges to indicate that the registrations belong to the same UA.
- Route - The target proxy for the message. The UA uses route headers to define the separate paths to the Oracle Enterprise Session Border Controller.

The Avaya SM uses reg-id in conjunction with instance-ID to manage dual registrations. By keeping instance-ID the same and sending a new reg-id, the infrastructure recognizes that a redundant registration was generated because a session manager switchover occurred.

Normally, multiple reg-ids based on a single contact would trigger a "move" procedure. The presence of a single instance-ID tells the infrastructure that the reg-id change does not indicate a move.

The following example REGISTERs depict the population of these elements for the purposes of an Avaya dual registration scenario.

```
 REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.2;branch=z9hG4bK-bad0ce-11-1036
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=d879h76
To: Bob <sip:bob@example.com>
Call-ID: 8921348ju72je840.204
Supported: path, outbound
Route: <sip:ep1.example.com;lr>
CSeq: 1 REGISTER Supported: path, outbound
Contact: <sip:line1@192.0.2.2;transport=tcp>; reg-id=1;
;+sip.instance="<urn:uuid:00000000-0000-1000-8000-000A95A0E128>" Content-
Length: 0
```

Note the following redundant registration. The registration includes a different route header for the second Oracle Enterprise Session Border Controller realm. It also includes a new reg-id and the same instance-ID.

```
 REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.2;branch=z9hG4bK-bad0ce-11-1036
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=d879h76
To: Bob <sip:bob@example.com>
Call-ID: 8921348ju72je840.204
Supported: path, outbound
Route: <sip:ep2.example.com;lr>
CSeq: 1 REGISTER Supported: path, outbound
Contact: <sip:line1@192.0.2.2;transport=tcp>; reg-id=2;
;+sip.instance="<urn:uuid:00000000-0000-1000-8000-000A95A0E128>" Content-
Length: 0
```

Registration Caching

Enabling the RFC 5626 SPL option causes the Oracle Enterprise Session Border Controller to store a single, entire contact header in its registration cache for the AOR. When an Avaya SM switchover occurs, the Oracle Enterprise Session Border Controller updates the AOR by replacing the contact header with the new one. The Oracle Enterprise Session Border Controller does not store more than one contact per AOR. The Oracle Enterprise Session Border Controller establishes a flow with only the active Avaya SM.

Configure Avaya Session Manager (SM) Redundancy

The rfc5636 SPL option allows the Oracle Enterprise Session Border Controller to support Avaya Dual Registration for establishing redundant UA registration.

The rfc5626 option must be configured at the global level under spl-config or using the Web GUI. The rfc5626 option is not recognized in the session-agent, realm-config, or sip-interface.

To configure the rfc5626 option:

1. In Superuser mode, type configure terminal and press <Enter>.
   ACMEPACKET# configure terminal
2. Type system and press <Enter>.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system and press <Enter>.

```
ACMEPACKET(system)# spl-config
ACMEPACKET(spl-config)#
```

4. Type rfc5626 and press <Enter>.

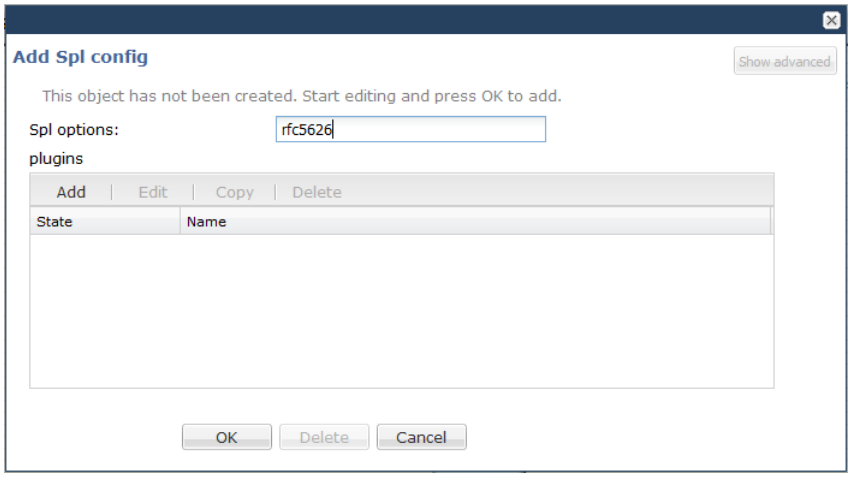
```
ACMEPACKET(spl-config)# rfc5626
ACMEPACKET(spl-config)#
```

5. Type done to save your work.
6. Save and Activate your configuration.

The following example shows an rfc5626 SPL configuration:

```
system
    spl-config
        spl-options           rfc5626
```

You can also configure the rfc5626 option using the Web GUI. The procedure consists of simply opening the spl-config dialog, adding the SPL option, then saving and activating.



27

P-Certificate-Subject-Common-Name to REGISTER Messages

Most Enterprises use revocation servers to authenticate certificates when user equipment registers with the Oracle Enterprise Session Border Controller. For high security enterprises, such as government organizations, user equipment, such as a cell phone, may have a certificate installed. If the user equipment is stolen, for example, the thief could use the equipment to register with theOracle Enterprise Session Border Controller and logon to the system before the certificate is revoked from the server.

The Oracle Enterprise Session Border Controller allows you to enable or disable the addition of a User certificate in the incoming REGISTER message header. This provides an additional layer of security when the user equipment registers with the Oracle Enterprise Session Border Controller. When the feature is enabled, the individual user certificate must match the user's identity during Registration.

You can enable or disable this feature using the "verify-certificate-info-register" parameter under the existing enforcement-profile object in session-router. in the ACLI. When enabled, and a REGISTER message is encountered, the Oracle Enterprise Session Border Controller adds the User certificate information to the message header. The header is then used in validating the Request-URI Based on certificate information.

Configure the P-Certificate-Subject-Common-Name From the ACLI

Use the following procedure to configure the P-Certificate-Subject-Common-Name.

To configure the P-Certificate-Subject-Common-Name:

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type enforcement-profile, and press Enter.

```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```

4. add-certificate-info—Enter sub-common name for the certificate attribute names to enable TLS certificate information caching, and for the inserting of cached certificate information into customized SIP INVITEs. Default is blank. Valid values are:
   - sub-common name
   - sub-alt-name-DNS
5. certificate-ruri-check—Enable this parameter if you want your Oracle Enterprise Session Border Controller to cache TLS certificate information and use it to validate Request-URIs. Enabling this parameter allows the Net-Net ESD to cache the TLS certificate information in a customized SIP INVITE. Default is disabled. Valid values are:
   - enabled
   - disabled
6. verify-certificate-info-register —Select whether or not to allow the Oracle Enterprise Session Border Controller to add certificate information to the header of a REGISTER message for verifying a ruri against certificate attributes. Default is disabled. Valid values are:
   - enabled
   - disabled
7. Type done, and press Enter.

```
ACMEPACKET(enforcement-profile)# done
ACMEPACKET(enforcement-profile)#
```

8. Type exit, and press Enter.

```
ACMEPACKET(enforcement-profile)# exit
ACMEPACKET(session-router)#
```

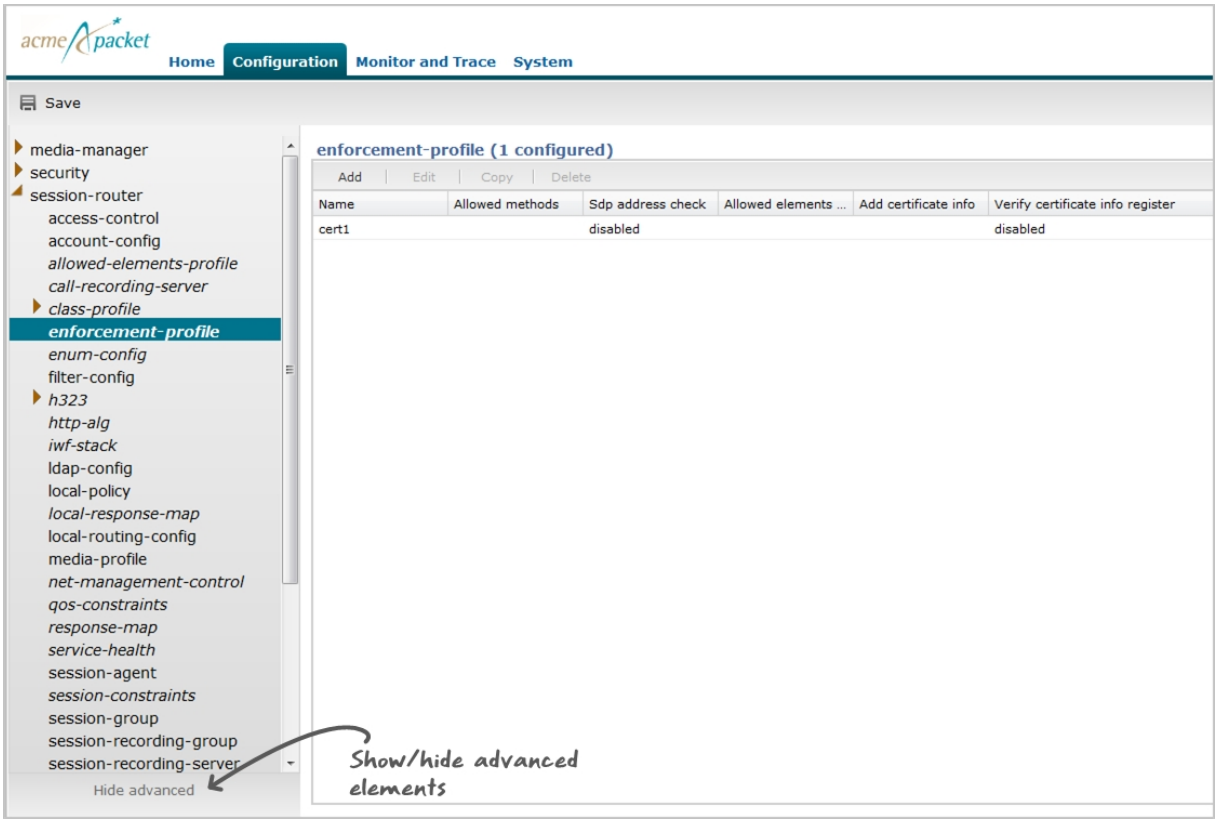
9. Save the configuration.

Configure the P-Certificate-Subject-Common-Name From the Web GUI

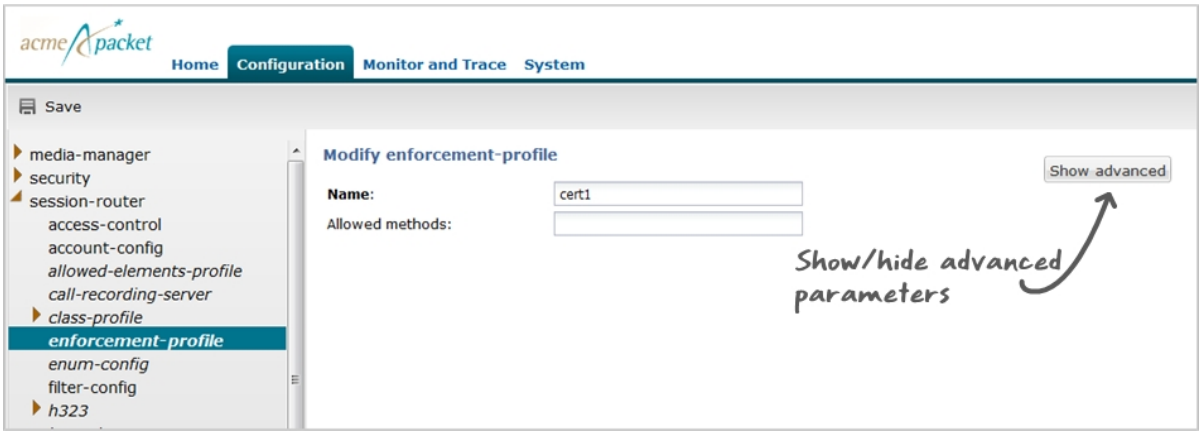
Use the following procedure to configure the P-Certificate-Subject-Common-Name using the Oracle Enterprise Session Border Controller Web GUI. In the Web GUI, this feature can be configured using Expert mode only and is an advanced configuration parameter.

To configure the P-Certificate-Subject-Common-Name in Expert mode:

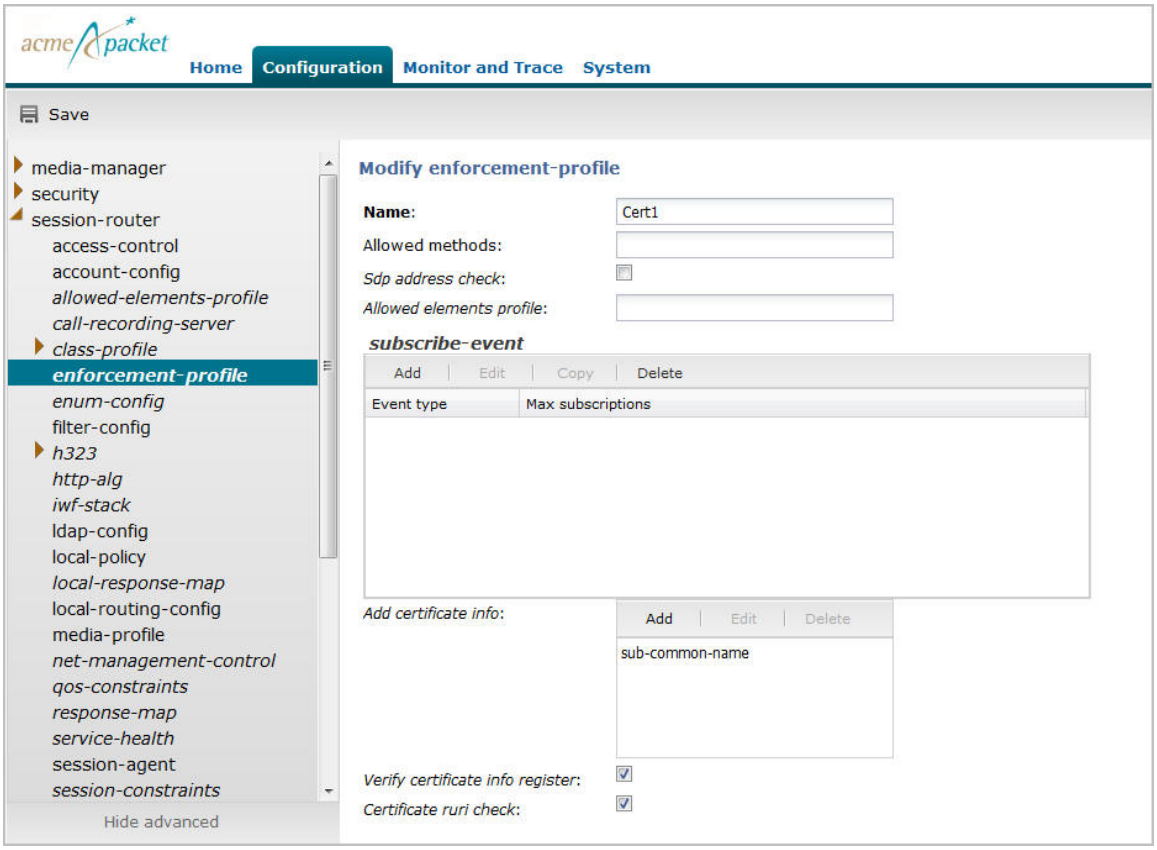
1. Logon to the Web GUI, and click Switch to Expert.
2. At the bottom of the left column, click Show advanced. The advanced elements for the objects in the left column display.



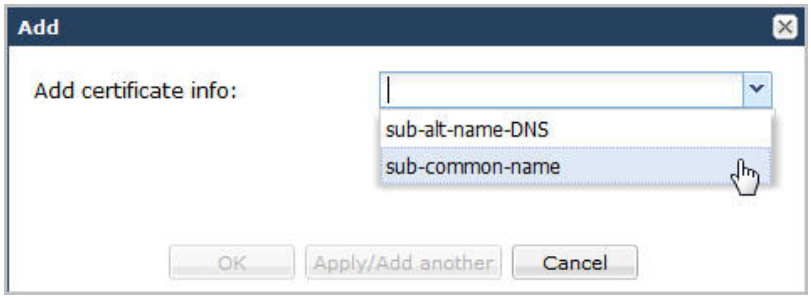
3. Click session-router.
4. Click enforcement-profile.
5. In the enforcement-profile dialog box, select the name of the certificate for which you want to enable the P-Certificate-Subject-Common-Name, and click <Edit>. The following dialog box displays.



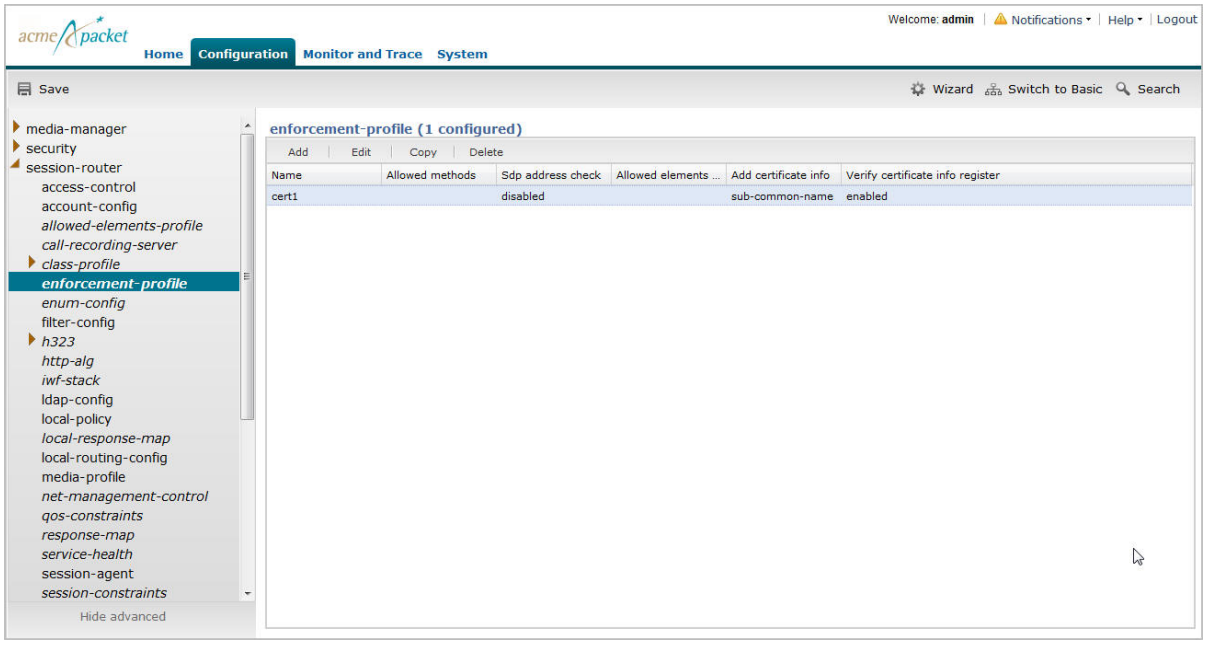
6. Click <Show advanced>. The advanced parameters for the certificate display.



7. In the Add certificate info box, click <Add>. The following dialog box displays.



8. Select sub-common-name from the drop-down box, and click <OK>.
9. In the Verify certificate info register field, place a check mark in the box to enable the Oracle Enterprise Session Border Controller to add certificate information to the header of a REGISTER message for verifying a ruri against certificate attributes. Click <OK>.
10. In the Certificate ruri check field, place a check mark in the box to enable this parameter if you want your Oracle Enterprise Session Border Controller to cache TLS certificate information and use it to validate Request-URIs. Enabling this parameter allows the Net- Net ESD to cache the TLS certificate information in a customized SIP INVITE. Click <OK>. The following window displays.



The certificate name has verify certificate info register enabled. The Oracle Enterprise Session Border Controller will include the sub-common name in the REGISTER message header before the UE registers.

28

SIP Monitor & Trace Enhancements

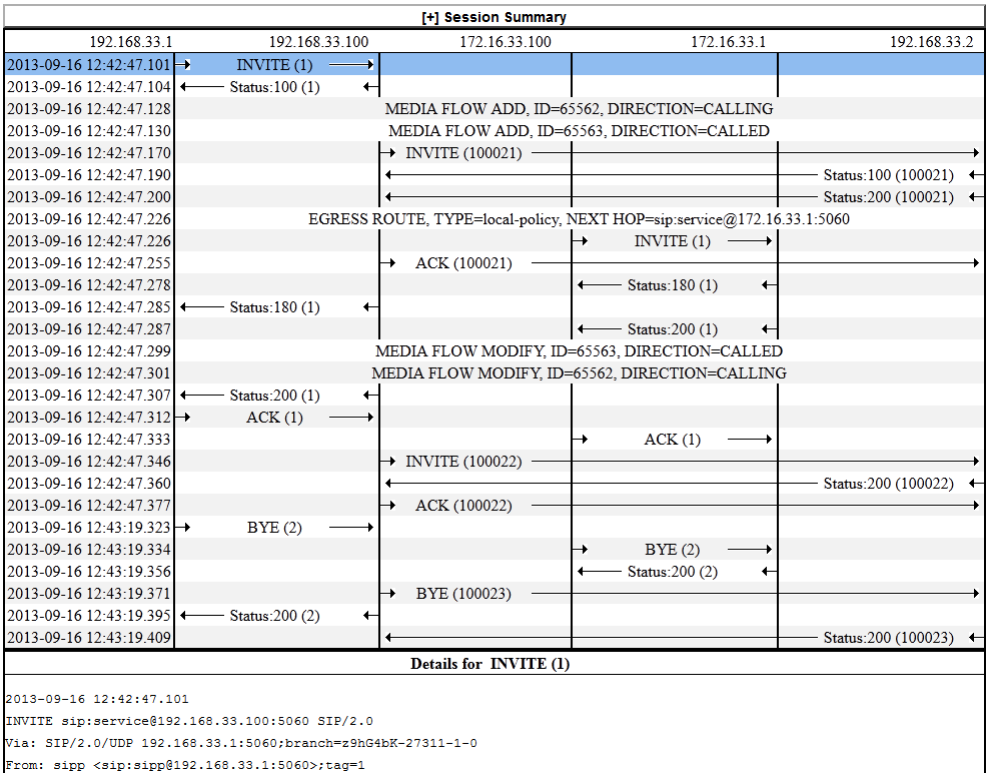
Release E-C[xz]6.4.0 M2 includes the following new feature enhancements for SIP Monitor and Trace (SM&T) in the Web GUI:

- SIPREC call data now captured and displayed in the ladder diagram in the Web GUI
- Hairpin call data now captured and displayed in the ladder diagram in the Web GUI
- SIP Monitor and Trace data now handled more efficiently on the Oracle Enterprise Session Border Controller

The following paragraphs describe these features.

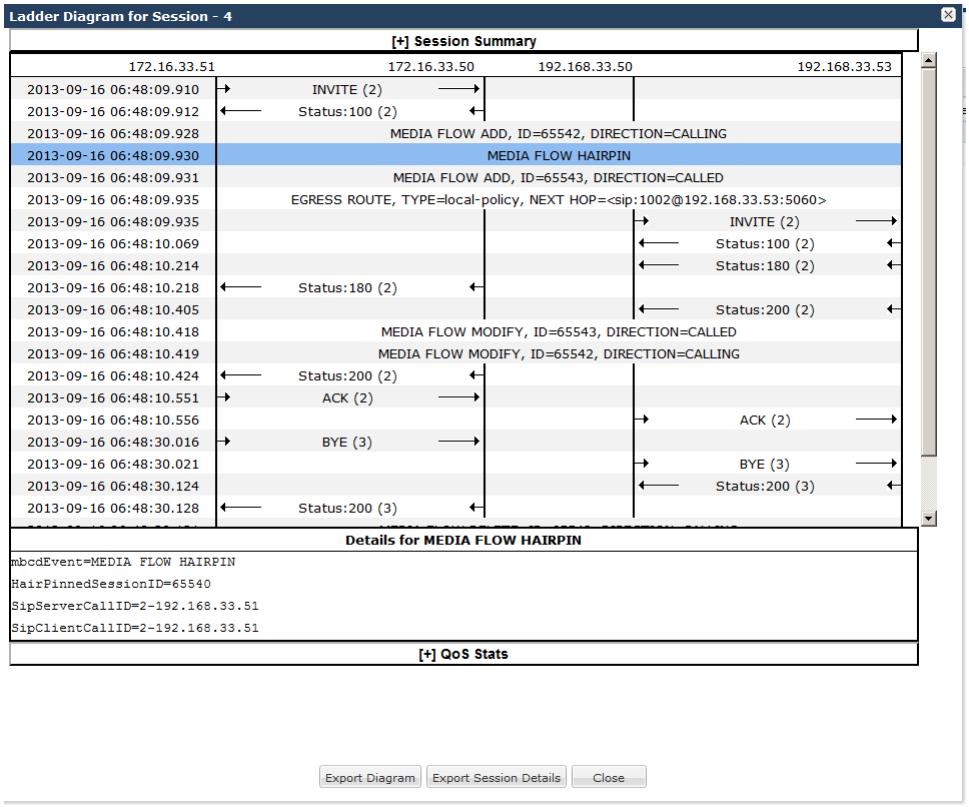
SIPREC Call Data

The following diagram shows SIP Monitor and Trace output for a call with media forwarded by way of SIPREC.



Hairpin Call Data

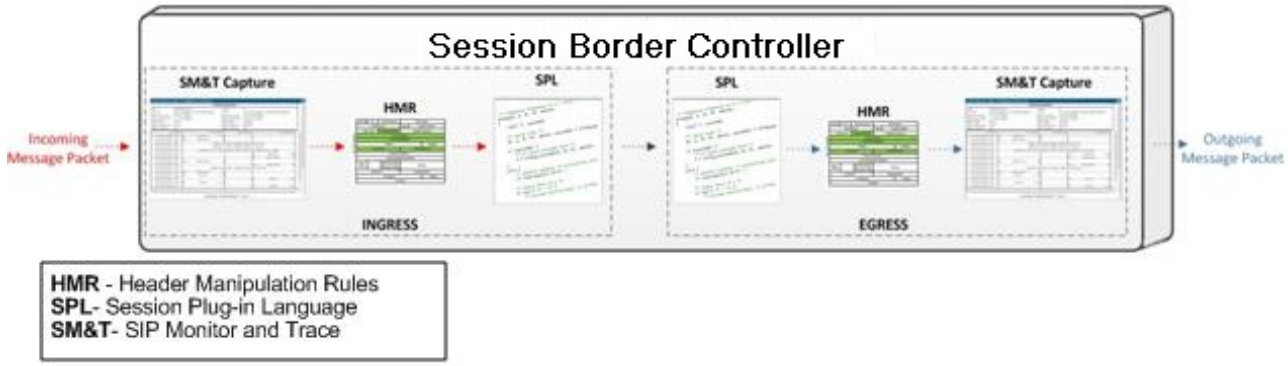
The following diagram shows SIP Monitor and Trace output for a hairpin call. Note the Media Flow Hairpin indication within the display.



SIP Monitor & Trace Ingress Egress Messages

The SM&T feature allows SIP sessions on the Oracle Enterprise Session Border Controller in your network to be monitored. Release E-C[xz]6.4.0 M2 includes a change to the way the Oracle Enterprise Session Border Controller handles SM&T data in ingress and egress messages. It processes SM&T data first on incoming messages and sends the data out last on outgoing messages. This allows the Oracle Enterprise Session Border Controller to capture SIP Monitor and Trace data over the wire for display in the Web GUI.

The Oracle Enterprise Session Border Controller captures SIP messages, applies the Header Manipulation Rules (HMR) configured on the Oracle Enterprise Session Border Controller, and then applies the Session Plug-in Language (SPL) to that message. When the message is sent out from the Net-Net ESD, it applies the SPL, then applies the HMR, and then sends out the captured SIP message.


Session Border Controller
SM&T Capture
HMR
SPL
SPL
HMR
SM&T Capture
Incoming
Message Packet
Outgoing
Message Packet
INGRESS
EGRESS
HMR - Header Manipulation Rules
SPL- Session Plug-in Language
SM&T- SIP Monitor and Trace

29

Web Server TLS Configuration and Management Commands

Introduction

You can configure Transport Layer Security (TLS) on the Web Server for enhanced security when accessing the server. You can also manage the server using specific management commands.

This chapter provides information about configuring TLS and using management commands on the Web Server.

Configuring TLS on the Web Server

The Web GUI supports the use of HTTP over Transport Layer Security (TLS) using the TLS Protocol. TLS is a cryptographic protocol that provides communication security over the Internet. It encrypts the segments of network connections at the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for privacy, and message authentication codes for message integrity.

Note: For more information about setting up security on your Net-Net ESD, see the chapter on security in this guide.

To use TLS with SIP Monitor and Trace, you must configure a TLS certificate and a TLS profile using the ACLI at the path **Configure Terminal** > **Security**. This configuration stores the information required to run SIP over TLS.

If you enable TLS on the active Net-Net ESD, the Web-based GUI interface on the standby system is disabled.

Process Overview

In summary, you need to take the following steps to enable your Net-Net ESD for TLS.

1. Make sure that your Net-Net ESD has the appropriate hardware installed and that you have obtained an enabled the licenses related to TLS support. (Note that the Net-Net 4250 does not require an additional license for TLS support.)
2. Configure certificates.
3. Configure the specific parameters related to TLS.

Configuring Certificates

Configuring certificates is a three-step process:

1. Create a certificate record configuration on the Net-Net ESD
2. Generate a certificate request by the Net-Net ESD and save the configuration
3. Import the certificate record into the Net-Net ESD and save the configuration

Configuring the Certificate Record

The certificate record configuration represents either the end-entity or the Certificate Authority (CA) certificate on the Net-Net ESD. If it is used to present an end-entity certificate, a private key should be associated with this certificate record configuration using the ACLI security certificate request command.

No private key should be associated with the certificate record configuration if it was issued to hold a CA certificate. A certificate can be imported to a certificate record configuration using the ACLI security certificate import command.

Note: There is no need to create a certificate record when importing a CA certificate or certificate in pkcs12 format.

To configure a certificate:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type security and press Enter to access the security-related objects.

   ```
   ACMEPACKET(configure)# security
   ```

3. Type certificate-record and press Enter to access the certificate record parameters.

   ```
   ACMEPACKET(security)# certificate-record
   ACMEPACKET(certificate-record)#
   ```

   name—Enter the name of the certificate record. This is a key field, and you must enter a value for it. For example, acmepacket.

   country—Enter the name of the country. The default is US.

   state—Enter the name of the state of for the country. The default is MA.

   locality—Enter the name of the locality for the state. The default is Burlington.

   organization—Enter the name of the organization holding the certificate. The default is Engineering.

   unit—Enter the name of the unit for the holding the certificate within the organization.

   common-name—Enter the common name for the certificate record.

   key-size—Enter the size of the key for the certificate. Use the default of 1024, or change it to one of the other supported values: 512, 2048, or 4096.

   alternate-name—Enter the alternate name of the certificate holder.

   key-usage-list—Enter the usage extensions you want to use with this certificate record. This parameter can be configured with multiple values, and it defaults to the combination of digitalSignature and keyEncipherment. For a list of possible values and their descriptions, see the section Key Usage Control in this guide.

   extended-key-usage-list—Enter the extended key usage extensions you want to use with this certificate record. The default is serverAuth. For a list of possible values and their descriptions, see the section Key Usage Control in this guide..

4. Enter done to save the certificate-record configuration.

   ```
   ACMEPACKET(certificate-record)# done
   ```

5. Enter exit to exit the certificate-record configuration.

   ```
   ACMEPACKET(certificate-record)# exit
   ```

6. Enter y at the prompt to save the configuration.

   ```
   Save Changes [y|n]?: y
   ```

7. Enter exit to exit the security configuration.

```
ACMEPACKET(security)# exit
```

8. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

9. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

10. Enter activate-config to activate as the current configuration.

```
ACMEPACKET# activate-config
```

Note: For verifying a certificate record, see the Security section of the Net-Net ACLI Configuration Guide for your Net-Net ESD model.

Generating a Certificate Request

Using the ACLI security certificate request command allows you to generate a private key and a certificate request in PKCS10 PEM format.

Note: You can only perform this step once you have configured a certificate record.

The Net-Net ESD stores the private key that is generated in the certificate record configuration in 3DES encrypted form with an internally generated password. The PKCS10 request is displayed on the screen in PEM (Base64) form.

You use this command for certificate record configurations that hold end-entity certificates. If you have configured the certificate record to hold a CA certificate, then you do not need to generate a certificate request because the CA publishes its certificate in the public domain. You import a CA certificate by using the ACLI security certificate request import command.

This command sends information to the CA to generate the certificate, but you cannot have Internet connectivity from the Net-Net ESD to the Internet. You can access the internet through a browser such as Internet Explorer if it is available, or you can save the certificate request to a disk and then submit it to the CA.

To run the applicable command, you must use the value you entered in the name parameter of the certificate record configuration. You run the command from the main Superuser mode command line, and then save and activate the configuration.

```
ACMEPACKET# security certificate request acmepacket
Generating Certificate Signing Request. This can take several
minutes....

-----BEGIN CERTIFICATE REQUEST-----

MIIB2jCCAUMCAQAwYTELMAkGA1UEBhMCdXMxCzAJBgNVBAgTAk1BMRMwEQYDVQQH
EwpCdXJsaW5ndG9uMRQwEgYDVQQKEwtFbmdpbmVlcmluZzEMMAoGA1UECxMDYWJj
MQwwCgYDVQQDEwNhYmMwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALOMLHo8
/qIOddIDVuqot0Y72l/BfH8lolRKmhZQ4e7sS+zZHzbG8phzmzhfOSECnZiA2bEo
f+Nti7e7Uof4lLwiYl9fvhURfzhENOKThAPKPiJCzBBglTITHTYal00Cq2fj5A8B
ZcuAHj7Vp5wP2zpz6EUTFpqTDMLVdwJGJrElAgMBAAGgOTAMBgNVHRExBRMDZGVm
MCkGA1UdDzEiEyBkaWdpdGFsU2lnbmF0dXJlLGtleUVuY2lwaGVybWVudDANBgkq
hkiG9w0BAQUFAAOBgQAtel4ZSLI8gqgMzodbYwgUHUGqTGeDzQDhJV5fKUXWeMFz
JsTmWn5Gy/kR4+Nq274G14fnk00fTAfMtgQ5aL3gM43TqaPOTZjJ6qgwuRKhoBPI
7hkovkgAxHge7wClghiAp/ELdl7tQ515k04BMd5f/fxG7nNiu8iEg7PO0OIBgg==
-----END CERTIFICATE REQUEST-----
WARNING: Configuration changed, run "save-config" command.
ACMEPACKET# save config
copying file /code/config/dataDoc.gz -> /code/config/dataDoc_3.gz
copying file /code/config/tmp/editing/dataDoc.gz ->
/code/config/dataDoc.gz
Save complete
ACMEPACKET# activate config
activate complete
```

Importing a Certificate Using the ACLI

For an end-entity certificate, once a certificate is generated using the ACLI security certificate request command, that request should be submitted to a CA for generation of a certificate in PKCS7 or X509v3 format. When the certificate has been generated, you can import it into the Net-Net ESD using the security certificate import command.

The syntax is:

```
ACMEPACKET # security certificate import [try-all | pkcs7 | pkcs12 |
x509] [certificate-record file-name]
```

To import a certificate:

1. When you use the security certificate import command, you can specify whether you want to use PKCS7, PKCS12, X509v3 format, or try all. In the command line, you enter the command, the format specification, and the name of the certificate record. Next, the Net-Net ESD will prompt you to enter the certificate in PEM format. Paste the certificate in the ACLI at this point. For example:

```
ACMEPACKET# security certificate import try-all acmepacket
The following displays:
Please enter the certificate in the PEM format.
Terminate the certificate with ";" to exit.......
-----BEGIN CERTIFICATE----
VMIIDHzCCAoigAwIBAgIIAhMCUACEAHEwDQYJKoZIhvcNAQEFBQAwcDELMAkGA1UE
BhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMQ4w
DAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lwaXQgVGVzdCBDZXJ0aWZpY2F0ZSBB
dXRob3JpdHkwHhcNMDUwNDEzMjEzNzQzWhcNMDgwNDEyMjEzNzQzWjBUMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCTUExEzARBgNVBAcTCkJ1cmxpbmd0b24xFDASBgNV
BAoTC0VuZ2luZWVyaW5nMQ0wCwYDVQQDEwRhY21lMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCXjIeOyFKAUB3rKkKK/+59LT+rlGuW7Lgc1V6+hfTSr0co+ZsQ
bHFUWAA15qXUUBTLJG13QN5VfG96f7gGAbWayfOS9Uymold3JPCUDoGgb2E7m8iu
vtq7gwjSeKNXAw/y7yWy/c04FmUD2U0pZX0CNIR3Mns5OAxQmq0bNYDhawIDAQAB
o4HdMIHaMBEGA1UdEQQKMAiCBnBrdW1hcjAJBgNVHRMEAjAAMB0GA1UdDgQWBBTG
tpodxa6Kmmn04L3Kg62t8BZJHTCBmgYDVR0jBIGSMIGPgBRrRhcU6pR2JYBUbhNU
2qHjVBShtqF0pHIwcDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEx
ETAPBgNVBAcTCFNhbiBKb3NlMQ4wDAYDVQQKEwVzaXBpdDEpMCcGA1UECxMgU2lw
aXQgVGVzdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHmCAQAwDQYJKoZIhvcNAQEFBQAD
gYEAbEs8nUCi+cA2hC/lM49Sitvh8QmpL81KONApsoC4Em24L+DZwz3uInoWjbjJ
QhefcUfteNYkbuMH7LAK0hnDPvW+St4rQGVK6LJhZj7/yeLXmYWIPUY3Ux4OGVrd
2UgV/B2SOqH9Nf+FQ+mNZOlL7EuF4IxSz9/69LuYlXqKsG4=
-----END CERTIFICATE-----;
Certificate imported successfully....
WARNING: Configuration changed, run "save-config" command.
```

2. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
copying file /code/config/dataDoc.gz -> /code/config/dataDoc_3.gz
copying file /code/config/tmp/editing/dataDoc.gz ->
/code/config/dataDoc.gz
Save complete
```

3. Enter activate-config to activate as the current configuration.

```
ACMEPACKET# activate-config
activate complete
```

Note: For importing a certificate using FTP, see the Security section of the *Net-Net ACLI Configuration Guide* for your Net-Net ESD model.

Importing a Certificate Using FTP

You can also put the certificate file in the directory /ramdrv and then execute the import-certificate command, or you can paste the certificate in PEM/Base64 format into the ACLI. If you paste the certificate, you may have to copy and paste it a portion at a time, rather than pasting the whole certificate at once.

To import the certificate using FTP:

1. FTP the certificate file on to the Net-Net ESD (directory /ramdrv), let us say the name of the certificate file is cert.pem.
2. Once the certificate is successfully transferred to the Net-Net ESD, run the import-certificate command.

   The syntax is:

   ```
   ACMEPACKET# import-certificate [try-all|pkcs7|x509] [certificate-record
   file-name]
   ```

   Using the command will look like this when you have used FTP.

   ```
   ACMEPACKET# import-certificate try-all acme cert.pem
   Certificate imported successfully....
   WARNING: Configuration changed, run "save-config" command.
   ```
3. Save your configuration.

   ```
   ACMEPACKET# save-config
   Save-Config received, processing.
   waiting 1200 for request to finish
   Request to 'SAVE-CONFIG' has Finished,
   Save complete
   Currently active and saved configurations do not match!
   To sync & activate, run 'activate-config' or 'reboot activate'.
   ```
4. Synchronize and activate your configurations.

   ```
   ACMEPACKET# activate-config
   Activate-Config received, processing.
   waiting 120000 for request to finish
   Add LI Flows
   LiSysClientMgr::handleNotifyReq
   H323 Active Stack Cnt:  0
   Request to 'ACTIVATE-CONFIG' has Finished,
   Activate Complete
   ACMEPACKET#
   ```

Configuring a TLS Profile

To configure a TLS profile:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type security and press Enter to access the security-related objects.

   ```
   ACMEPACKET(configure)# security
   ```
3. Type tls-profile and press Enter to access the TLS profile-related parameters.

   ```
   ACMEPACKET(security)# tls-profile
   ACMEPACKET(tls-profile)#
   ```

   name—Enter the name of the TLS profile. This parameter is required; you cannot leave it empty.

   ```
   ACMEPACKET(tls-profile)# name tls-prof1
   ```

   end-entity-certificate—Enter the name of the entity certification record.

   ```
   ACMEPACKET(tls-profile)# end-entity-certificate cert1
   ```

   trusted-ca-certificates—Enter the names of the trusted CA certificate records.

   ```
   ACMEPACKET(tls-profile)# trusted-ca-certificates cert1
   ```

   Note: To create and import certificate records to be used on the Web Server, see Configuring Certificates.

   cipher-list—Not supported for SIP Monitor and Trace. The Session Director ignores any value you enter for this parameter.

- AES256-SHA (TLS_RSA_WITH_AES_256_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m) only
- AES128-SHA (TLS_RSA_WITH_AES_128_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m) only
- DES-CBC-SHA (SSL_RSA_WITH_DES_CBC_SHA or TLS_RSA_WITH_DES_CBC_SHA) - Internet Explorer (Version 9) only

verify-depth—Not supported for SIP Monitor and Trace

mutual-authenticate—Not supported for SIP Monitor and Trace

tls-version—Enter the TLS version you want to use with this TLS profile. Default is compatibility. Valid values are:

- TLSv1
- SSLv3
- compatibility (default)

```
ACMEPACKET(tls-profile)# tls-version TLSv1
```

cert-status-check—Not supported for SIP Monitor and Trace

cert-status-profile-list—Not supported for SIP Monitor and Trace

ignore-dead-responder—Not supported for SIP Monitor and Trace

allow-self-signed-cert—Not supported for SIP Monitor and Trace

4. Enter done to save the tls-profile configuration.

```
ACMEPACKET(tls-profile)# done
```

5. Enter exit to exit the TLS profile configuration.

```
ACMEPACKET(tls-profile)# exit
```

6. Enter exit to exit the security configuration.

```
ACMEPACKET(security)# exit
ACMEPACKET(configure)#
```

7. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

8. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

9. Enter activate-config to activate as the current configuration.

```
ACMEPACKET# activate-config
```

Management Commands for the Web Server

The following commands allow you to display information for managing the Web server used for accessing the GUI.

Command	Description
show ip connections	Displays information about the server connections.
show users	Displays information about users logged into a session on the server.
kill <index>	Terminates a session on the server.

Show ip connections Command

The show ip connections command allows you to display information about active server Transport Control Protocol (TCP) and/or User Datagram Protocol (UDP) connections. For example, this command can show the sockets tied to an HTTPS connection. The following is an example of the show ip connections command output.

```
ACMEPACKET# show ip connections
Active Internet connections (including servers)
PCB      Proto Recv-Q Send-Q  Local Address          Foreign Address
(state)
-------- ----- ------ ------  --------------------- ---------------------
-------
75059a0  TCP        0      0  172.30.80.231.1538    172.30.0.39.58497
TIME_WAIT
7506420  TCP        0      0  172.30.80.231.443     10.1.20.14.51006
TIME_WAIT
75044a0  TCP        0      0  172.30.80.231.443     10.1.20.14.51000
TIME_WAIT
7504f20  TCP        0      0  172.30.80.231.443     10.1.20.14.50997
TIME_WAIT
7503f60  TCP        0      0  127.0.0.1.3000        127.0.0.1.1064
ESTABLISHED
7503a20  TCP        0      0  127.0.0.1.3000        127.0.0.1.1063
ESTABLISHED
75034e0  TCP        0      0  127.0.0.1.3000        127.0.0.1.1062
ESTABLISHED
7502fa0  TCP        0      0  127.0.0.1.3000        127.0.0.1.1061
ESTABLISHED
7502a60  TCP        0      0  127.0.0.1.3000        127.0.0.1.1060
ESTABLISHED
7502520  TCP        0      0  127.0.0.1.1063        127.0.0.1.3000
ESTABLISHED
7501fe0  TCP        0      0  127.0.0.1.1062        127.0.0.1.3000
ESTABLISHED
7501aa0  TCP        0      0  127.0.0.1.1061        127.0.0.1.3000
ESTABLISHED
```

The following table describes each column in the above output.

Column Heading	Description
PCB	Printed circuit board in the server that is active on the connection.
Proto	Protocol used on this connection. Valid values are: TCP - Transport Control Protocol UDP - User Datagram Protocol
Recv-Q	Receiving queue - pertains to the queue on the server that receives packets from the Internet. This column should always display a zero (0). Packets should not be piling up in this queue.
Send-Q	Sending queue - pertains to the queue on the server that sends out packets to the Internet. This column should always display a zero (0). Packets should not be piling up in this queue.
Local Address	Local server's IP address and port number, or IP address and the name of a service.
Foreign Address	Hostname and service, or IP address and port number to which you are connected. The asterisk is a placeholder for IP addresses, which of course cannot be known until a remote host connects.
(state)	Current state of the TCP or UDP connection. TCP states can be:

Column Heading	Description
	LISTEN waiting to receive a connection ESTABLISHED a connection is active TIME_WAIT a recently terminated connection; this should last only a minute or two, then change back to LISTEN. The socket pair cannot be re-used as long the TIME_WAIT state persists. UDP is stateless, so the "State" column is always blank.

Show users Command

The show users command displays information about users currently logged into a session on the server. Each user is indicated by an index number. The following is an example of the show users command output.

Note: The index number for Web sessions always begins at 31.

```
ACMEPACKET# show users
Index task-id    remote-address         IdNum duration type    state     User
----- ---------- --------------------- ----- -------- ------- ------
----------
    0 0x33a4c394                            0 00:01:20 console  user * console
    1 0x33a68858     172.30.0.39:39385      1 00:00:08  telnet  user  user
   31     NA          10.1.20.14:51218     31 00:00:55    http  user  user
   32     NA            10.1.20.14:443     32 00:00:29   https  user  user
```

The following table describes each column in the above output.

Column Heading	Description
Index	Number that the server assigns to the user as an identification of that user. Note: The index for Web sessions always begins at index 31.
Task-id	Alpha-numeric number that the server assigns to the task currently being performed. This is the session ID assigned to the task at log-in time. This field is not applicable to the Web server.
Remote-address	IP address and port number for which the server is connected.
IdNum	Identification number of the user currently logged into the server. This number is the same as the Index number.
Duration	Amount of time, in hours, minutes, and seconds, that the user has currently been logged into a session on the server. Format is HH:MM:SS.
Type	Type of service that the user is currently using for connection to the server. Valid values can be: Console User is connected to the server via a local console. Telnet User is connected to the server via a Telnet session. FTP User is connected to the server using FTP. HTTP User is connected to the server using a Web HTTP service. HTTPS User is connected to the server using a secure Web HTTPS service.
State	Current state of the connection on the server. Valid values are: admin

Column Heading	Description
	user Note: An “*’ indicates a current connection.
User	Current user type logged into the server. Valid values are: console user admin

Kill <index> Command

The kill <index> command terminates a session on the server. The following example uses the “show users” command to display the index number to use with the kill <index> command.

```
ACMEPACKET# show users
Index task-id    remote-address         IdNum duration type    state     User
----- ---------- ---------------------- ----- -------- ------- ------
----------
    0 0x33a4c394                            0 00:01:20 console  user * console
    1 0x33a68858     172.30.0.39:39385      1 00:00:08  telnet  user  user
   31     NA          10.1.20.14:51218     31 00:00:55    http  user  user
   32     NA            10.1.20.14:443     32 00:00:29   https  user  user

ACMEPACKET# kill 31
```

The above kill 31 command terminates the Web server session number 31.

After setting the Web server configuration, you can view the stored monitored data from your Net-Net ESD(s) using the GUI via your Internet browser. For more information about the GUI, see Configuring TLS on the Web Server.

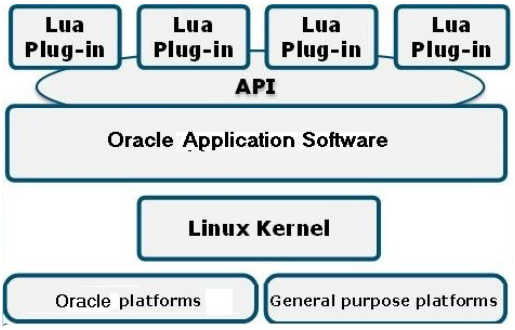
30

Session Plug-in Language (SPL)

Oracle SPL Plug-ins

An SPL is an Oracle signed plug-in that integrates with the Oracle Enterprise Session Border Controller (E-SBC) application software to quickly add feature extensions without requiring a software upgrade or causing operational impacts. Each SPL plug-in is an executable, customized script that is based on the Lua open scripting language. Oracle SPL plug-ins allow you to add enhancements when you need them, rather than waiting for the next software release.

The following illustration shows how an SPL plug-in integrates with the E-SBC platform.



The system includes the SPL engine that runs the SPL plug-in scripts and each release supports a number of versions of the SPL engine. For a list of supported versions, see "System Programming Language (SPL) Engine Support".

The SPL can run on any platform with an ANSI C compiler. You can use the SPL with Header Manipulation Rules by way of the E-SBC ACLI configuration.

General SPL Information

This section describes the general information about SPLs. This information pertains to all the SPLs described in this chapter, unless otherwise specified for the respective SPL.

Supported Platforms

The following Oracle Enterprise Session Border Controller platforms support using SPL plug-ins:

- Acme Packet 4500

- Acme Packet 3820
- Server Edition (certified servers include HP DL120 G7, HP DL320e G8, and Dell R210 II)
- VMWare (preferred Hypervisor)

Load and Enable an SPL Plug-in

Some System Programming Language (SPL) plug-ins require manual loading onto the Oracle Enterprise Session Border Controller (E-SBC.

The process to load and enable an SPL includes the following steps.

1. Upload the SPL to the E-SBC. See "Upload an SPL Plug-in."
2. Add the SPL to the E-SBC configuration. See "Add an SPL Plug-in to your Configuration."
3. In a High Availability (HA) deployment, synchronize the SPL files across the HA pair. See "Synchronize an SPL Plug-in File Across an HA Pair."

Upload an SPL Plug-in

Manually FTP the SPL plug-in to the /code/spl directory on the Oracle Enterprise Session Border Controller (E-SBC) using any CLI or GUI-based FTP or SFTP application. Use the wancom or eth0 management physical interface on the E-SBC to reach the FTP or SFTP server.

Add an SPL Plug-in to the Configuration

You must add an SBC Programming Language (SPL) plug-in file to the spl-config element before the system can execute the plug-in. The system ignores any SPL plug-in that exists in the /code/spl directory that is not included in the spl-config element.

Before You Begin

- Confirm that you have Superuser permissions

In the following procedure, you add the name of one or more SPL plug-ins to the spl-config configuration element. Note that the system executes SPL plug-ins in the order in which they were configured.

Procedure

1. Access the **spl-config** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# spl-config
ACMEPACKET(spl-config)#
```

2. Type plugins , and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMESYSTEM(spl-config)# plugins
ACMESYSTEM(spl-plugins)#
```

3. Type name, a space, and the name of the SPL file.

```
ACMESYSTEM(spl-plugins)#name MediaPlayback.1.0.spl
ACMESYSTEM(spl-plugins)#name LyncEmergencyCall.1.0.spl
ACMESYSTEM(spl-plugins)#name SipHeaderExtensionMetadata.1.2.spl
ACMESYSTEM(spl-plugins)#name UniversalCallId.1.spl
ACMESYSTEM(spl-plugins)#name ComfortNoiseGeneration.1.1.spl
```

4. Type done to save your work.
5. Save and activate the configuration.

Next Steps

- If your deployment supports a High Availability (HA) pair configuration, see "Synchronize SPL Files Across HA Pairs."

Execute an SPL Plug-in File

After you add one or more SPL plug-in files to the configuration, save and activate the configuration to execute the plug-ins.

- From the command line, perform a save-config and activate-config after exiting the configuration menu.

If an SPL file exists in the /code/spl directory, but is not configured in the ACLI, it is ignored when the ACLI user interface is booting.

Synchronize SPL Plug-in Files Across an HA Pair

In a High Availability (HA) configuration, both the active and the standby systems require the same version of the SBC Programming Language (SPL) plug-in script.

There is no means to synchronize SPL files automatically during a save and activate after you add SPL files to the configuration. To configure the standby system, execute the synchronize spl ACLI command. Note that the system only executes the synchronize spl command from the active system in a HA pair.

To copy all files in the /code/spl directory from the active system to the same directory on the standby do not include any arguments. Note that this procedure overwrites any existing files on the standby system with the same name.

To copy individual files, add the specific filename as an argument to the synchronize spl command, for example,

```
ACMEPACKET#synchronize spl MediaPlayback.1.0.spl
ACMEPACKET#synchronize spl LyncEmergencyCall.1.0.spl
ACMEPACKET#synchronize spl SipHeaderExtensionMetadata.1.2.spl
ACMEPACKET#synchronize spl UniversalCallId.1.spl
ACMEPACKET#synchronize spl ComfortNoiseGeneration.1.1.spl
```

Procedure

- In Superuser mode, type synchronize spl, and press Enter.

```
ACMEPACKET# synchronize spl
```

Local Media Playback

Commonly, ringback is the media playback of a certain tone informing callers that their calls are in progress. In typical deployments, remote endpoints or media servers handle ringback generation, leaving the Oracle Enterprise Session Border Controller (E-SBC) to proxy RTP. When endpoints or media servers do not support ringback generation, the E-SBC becomes the producer.

Note: The E-SBC supports a maximum of 100 simultaneous play backs.

You can configure the E-SBC to generate ringback locally, meaning it can produce RTP media on a media flow. The most common use for enabling the system to produce RTP on a media flow is to support locally generated ringback. Because you can also use this capability for music-on-hold, announcements, and interrupting media for notifications, this E-SBC capability is referred to as local playback.

Local playback is controlled through the ACLI using the Local Media Playback SPL plug-in.

Supported Capabilities and Caveats

The Oracle Enterprise Session Border Controller (E-SBC) supports the following playback scenarios:

- Playback on 183 Session Progress
- Playback on REFER
- Playback on header, where the header is P-Acme-Playback

The system supports local media playback for the following E-SBC capabilities:

- SRTP
- Call recording

- SIPREC

Local playback does not work in call flows for which media is released. Concurrent playbacks are limited to 100.

Pre-Requisites

The Local Playback SPL plug-in is installed in the software, and you must fulfill the following requirements for Local Playback to work properly.

1. Configure one or more playback configuration elements. See, *Setting up the Playback Configuration*.
2. Configure the realm, session agent, or SIP interface objects with the necessary SPL playback option(s) for your deployment. See *Setting Playback Options on Realms Session Agents and SIP Interfaces*.

Media Files

You must upload a file containing to /code/media to the E-SBC for the media that you want played. This file must be raw media binary containing data for the desired codec. A separate file is required for each different codec type, even if the media itself is the same.

Your configuration must specify a playback rate in bytes per second, as this setting defines how many bytes of data per unit of ptime are needed.

To preserve system memory resources, media files are limited to 2MB.

ACLI Configuration and Examples

Use the following procedure to configure the Local Media Playback options on the Oracle Enterprise Session Border Controller.

1. Set up the playback configuration and associated entries sub-elements.
2. Set up the realm, session-agents, or sip-interfaces for which you want to enable local playback.

Set up the Playback Configuration

The playback configuration defines the media files that you want to play, listed by codec. Each codec encoding that you want to support requires its own file, defined by the playback entry sub-element.

Procedure

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager, and press Enter.

```
ACMEPACKET(configure)# media
ACMEPACKET(media-manager)#
```

3. Type playback-config , and press Enter.

```
ACMEPACKET(media-manager)# playback-config
ACMEPACKET(playback-config)#
```

4. name—Enter the name of this playback configuration. This parameter has no default, and is required. You use this name when you configure the spl-options parameter; it specifies the media the Oracle Enterprise Session Border Controller (E-SBC) plays.
5. entries—Configure the entries for this playback configuration. These entries refer to files in the /code/media directory, and are designed so that you can reference the same media with different codecs. For example, you might want to be able to play the same playback tone in different codecs; here, you would specify the file name and the encoding for each one you have stored.

```
ACMEPACKET (playback-config)# entries
ACMEPACKET(entries)#
```

   - encoding—Enter the codec name for this media file entry, such as PCMU. This value must match the encoding name negotiated in the SDP. This parameter has no default and is required.

- filename—Enter the name of the raw binary media file you stored in the /code/media directory on the E-SBC.
- bytes-per-second—Enter the playback rate for this media file in bytes per second. Default: to 8000. Range: 100-99999.

6. Type done, and save the configuration.

Playback Configuration Example

The following is an example of a playback configuration:

```
playpback-config
    name              Ringback
    entries
        encoding          PCMU
        filename          tonePCMU.rbf
        bytes-per-second  8000
    entries
        encoding          PCMA
        filename          tonePCMA.rbf
        bytes-per-second  8000
    entries
        encoding          G729
        filename          toneG729.rbf
        bytes-per-second  8000
```

Set Playback Options on Realms, Session Agents, and SIP Interfaces

To set playback options for realms, session agents, and SIP interfaces on the Oracle Enterprise Session Border Controller (E-SBC), you can set the spl-options parameter under each of those objects in the ACLI. The spl-options parameter allows you to set the following options:

- playback-on-183-to-originator—Playback enabled upon the receipt of a 183 Session Progress destined for the originator and stops when a either a 200-299 or 400-699 final response is sent.
- playback-on-183-from-terminator—Playback enabled upon the receipt of a 183 Session Progress response is received from the terminator and stops when a 200-299 or 400-699 final response is received.
- playback-on-refer—Playback enabled for the caller being transferred when the E-SBC receives a REFER message that is locally terminated (i.e., processed on the E-SBC on REFER completion).
- playback-on-header—Starts or stops playback based on the presence of the P-Acme-Playback header and its definitions.

Note: The E-SBC supports a maximum of 100 simultaneous playbacks.

With each of these options, you set the name of the playback configuration you want to use for the associated scenario. Except with playback-on-header, if there is one file to play, the E-SBC loops it continuously until a stop event occurs. If multiple media files can be used for the same events, the E-SBC queues them in the order you specify in your configuration and plays them consecutively until a stop event occurs. For playback-on-header, you can define how you want playback to work by setting the duration mode in the P-Acme-Playback header to once, continuous, or the amount of time in milliseconds you want the media to play.

The following example shows you how to configure the playback-on-refer SPL option for a session agent. The steps are the same for the spl-options parameter in realms, session agents, and SIP interfaces. Simply set the option you want in the configuration where it needs to be applied.

Procedure

1. Access the **session-agent** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)
```

2. Type spl-config , and press Enter.

```
ACMEPACKET(session-agent)# spl-config
ACMEPACKET(spl-config)#
```

3. Type spl-options, followed by the option name pre-pending with the plus sign (+). If you do not use the plus sign, the system overwrites any other options that you configured previously.

   ```
   ACMESYSTEM(spl-config)# spl-options +playback-on-refer=media1,media2
   ```

   In the previous example, the E-SBC will play media1 and then media2.

4. Type done and save the configuration.

Playback-on-Refer Configuration Example

The following code shows an example of a playback-on-refer configuration.

```
session-router
    session-agent
        spl-config
            spl-options          +playback-on-refer=media1,media2
```

RTC Support

The playback configuration is supported by real-time configuration (RTC). Media files located in the /code/media directory and referenced by playback configuration entries are loaded at boot time and when you activate a configuration. The system does not reload any media being played to an endpoint. Playbacks that start after the boot or configuration activation use updated media files.

Import and Export the E-SBC Configuration

You can import and export the Oracle Enterprise Session Border Controller (E-SBC) configuration to and from a Comma Separated Value (CSV) file with the Import Export SPL plug-in. The Import Export SPL plug-in is enabled by default. No configuration is required.

Import and Export Restrictions

The following table lists the restrictions that the system enforces when importing and exporting the Oracle Enterprise Session Border Controller (E-SBC) configuration.

Import and Export Restrictions
Files are read/written to the volatile directory of the file system on the E-SBC.
Import and export occurs to and from the editing configuration.
The system displays all error messages on the screen where the command was issued and provides line numbers with the error, when possible. The system does not allow you to set objects and attributes to inappropriate values. For example, you cannot set an IP address to "enabled". Parsing continues as normal after this error. If the system cannot write an object, for example, when the key field is missing, the system discards the object and parsing continues as normal. The import is additive. Each object that is imported is expected to be new to the configuration. If there is already an object with the same key present, the system generates the 409 error and discards the object. Parsing continues as normal after the error.

Import an E-SBC Configuration from a CSV File

The Oracle Enterprise Session Border Controller (E-SBC) import feature uses the import and export SPL plug-in. You must use a spreadsheet application, such as MS-Excel, that supports CSV files.

The following rules apply when entering configuration data into the CSV file.

CSV Configuration Import Rules
Empty lines are ignored
The first non-empty line must be the keyword “object:”, followed by the configuration object name that is being configured (shown below as sip-interface). `Example: object:sip-interface`
The second non-empty line must be the parameter names of the objects to be configured, each parameter name in its own column. This row defines the labels for each column for the subsequent rows. Only the attributes that you want defined need to be present. You can specify the parameter names in any order, but the data in subsequent rows must be consistent with the labels that you define in this row. `Example: state,realm-id,description`
The third non-empty rows define values for the configuration object, each instance in its own column. In the following example, the third line defines a new sip-interface with state “enabled”, realm-id “public”, and description “public SIP interface”. These values are based on the labels defined in the second row. `Example: enabled,public,public SIP interface`
On all subsequent rows, you can define any number of instances.

The next row with an “object” keyword selects a new configuration object that is based on the previous object. You continue to input the data for this object according to the rules stated above. The following example shows a “sip-port” object added that is related to the sip-interface object.
`Example: object:sip-port`
`address,port,transport-protocol`
`192.168.1.1,5060,UDP`
`192.168.1.1,5061,TCP`
In the example above, “sip-port” is a sub-object of “sip-interface” and would create new sip-ports off of the last sip-interface instance (of realm-id public).

Create a CSV File

To create a Oracle Enterprise Session Border Controller (E-SBC) configuration using a CSV file:

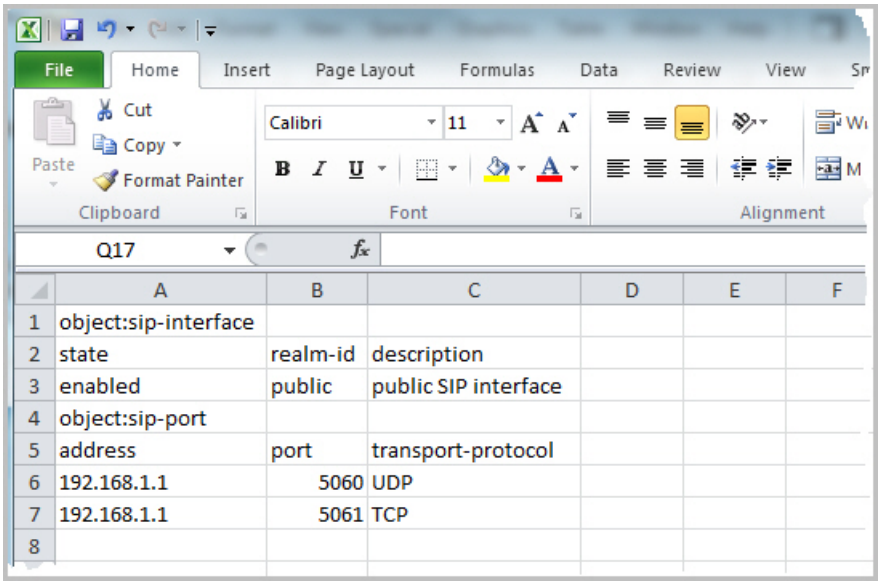
1. Open an application that supports a CSV file, for example, MS-Excel.
2. In the first row, first column, enter object: followed by a configuration object that you want to import.

   ```
   object:sip-interface
   ```
3. In the second row, and each in its own column, enter the parameter names of the objects to configure.

   ```
   state,realm-id,description
   ```
4. In the third row, and each in its own column, enter the values for the configuration objects.

   ```
   enabled,public,public SIP interface
   ```
5. In subsequent rows, define additional values, if any.
6. In the next empty row, first column, enter another object if required, related to the first object (sip-interface).

   ```
   object:sip-port
   ```
7. Repeat steps 3 through 5 for this object. The following is an example of a CSV file containing the “sip-interface” and sip-port objects.



	A	B	C	D	E	F
1	object:sip-interface					
2	state	realm-id	description			
3	enabled	public	public SIP interface			
4	object:sip-port					
5	address	port	transport-protocol			
6	192.168.1.1	5060	UDP			
7	192.168.1.1	5061	TCP			
8						

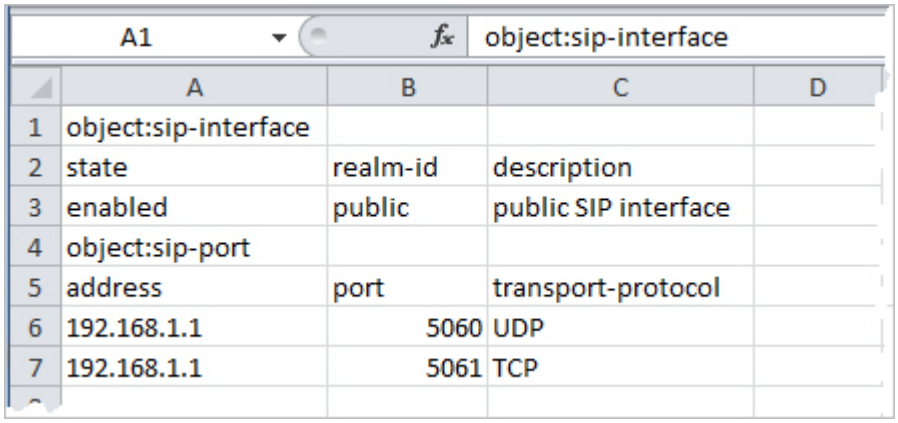
8. Save the file as a comma-separated value file (.csv). For example,

```
nnesd_config.csv
```

9. Close the file.

Enter Configuration Data Using a Text File

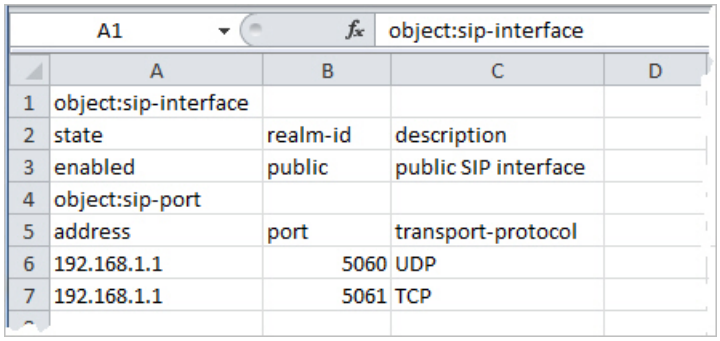
You can create a configuration by entering the required data into a text file and then use an application, such as Excel, to open and save the file as a CSV file. You must enter the information in the text file in the exact format as shown in the following example so that the labels to go to the correct columns in the Excel application.



	A	B	C	D
1	object:sip-interface			
2	state	realm-id	description	
3	enabled	public	public SIP interface	
4	object:sip-port			
5	address	port	transport-protocol	
6	192.168.1.1	5060	UDP	
7	192.168.1.1	5061	TCP	

Procedure

1. Enter the required configuration data in a text file according to the format in the previous illustration.
2. Save and close the file.
3. Open an application that supports a CSV file, for example, MS-Excel.
4. Browse for the SPL text file and open it. The configuration data opens in the correct columns within the application.



	A	B	C	D
1	object:sip-interface			
2	state	realm-id	description	
3	enabled	public	public SIP interface	
4	object:sip-port			
5	address	port	transport-protocol	
6	192.168.1.1	5060	UDP	
7	192.168.1.1	5061	TCP	

5. Save the file as a comma-separated value file (.csv). For example,

```
nnesd_config.csv
```

6. Close the file.

Import ACLI Command

After you create a CSV file that contains your Oracle Enterprise Session Border Controller (E-SBC) configuration, you can import the file into the E-SBC using the following command:

```
spl load acli config-csv <filename>
```

This command loads the CSV file from the volatile file system (/ramdrv/ for 4500), (/var/ for NN-ESD), into the editing configuration of the E-SBC.

Before You Begin

- You must have a CSV file already created that contains the new configuration data that you want to import.
- Initiate the delete-config cached command to clear the current editing configuration before importing to prevent conflicts.

Procedure

1. Access the E-SBC locally through the console connection or remotely through a TELNET or SSH connection.
2. Transfer the CSV file using FTP or SFTP to the volatile file system on the E-SBC (/ramdrv/for 4500), (/var/ for E-SBC).
3. At the prompt, type the applicable password for entering the ACLI, and press Enter. For example:

```
Password: acme
```

4. Enter enable to access the Superuser mode, followed by the password, and press Enter. For example:

```
NN-ESD> enable
Password: packet
NN-ESD#
```

Note: The passwords used above are the default passwords for the ACLI. These passwords may have been changed by your System Administrator. Contact your System Administrator for more information.

5. At the prompt, type delete-config cached to clear the current editing configuration on the E-SBC.

   NN-ESD# delete-config cached

6. Type spl load acli config-csv <filename>, and press Enter. For example:

   NN-ESD# spl load acli config-csv esd-config.csv

   The NN-ESD imports the CSV file, containing the configuration you specified, into the editing configuration of the E-SBC OS.

7. Type save, and press Enter.
8. Typeactivate, and press Enter. The imported configuration data becomes part of the editing configuration on the E-SBC.

Note: If you need to undo the import, initiate the restore-backup-config running command. This command restores the editing configuration that existed before you performed the import.

The following code block shows an example of the imported configuration.

```
sip-interface
    state enabled
    realm-id public
    description public SIP interface
    sip-port
        address 192.168.1.1
        port 5060
        transport-protocol UDP
    sip-port
        address 192.168.1.1
        port 5061
        transport-protocol TCP
```

Note: If you need to undo the import, initiate the restore-backup-config running command. This command restores the editing configuration that existed before you performed the import.

Export an E-SBC Configuration to a CSV File

You can export an existing configuration from the Oracle Enterprise Session Border Controller (E-SBC)to a CSV file using the following command:

```
spl save acli config-csv <filename>
```

This command saves the editing configuration on the E-SBC to a CSV file with the filename that you specify, and stores it on the volatile file system (/ramdrv/ for 4500), (/var/ for E-SBC).

Export ACLI Command

Use the following procedure to export the NN-ESD configuration into a CSV file that you specify.

Pre-requisite:

- The NN-ESD must have an editing configuration currently loaded to the system.
- If required, you can initiate the restore-backup-config running command to copy the current running configuration into the editing configuration.

To export the NN-ESD configuration to a CSV file:

1. Locally access the NN-ESD through the console connection or remotely via a TELNET or SSH connection.
2. At the prompt, enter the applicable password for entering the ACLI and press Enter. For example:

   ```
   Password: acme
   ```

3. Enter enable to access the Superuser mode, followed by the password and press Enter. For example:

   ```
   NN-ESD> enable
   Password: packet
   NN-ESD#
   ```

   Note: The passwords used above are the default passwords for the ACLI. These passwords may have been changed by your System Administrator. Contact your System Administrator for more information.

4. At the prompt, enter spl save acli config-csv <filename>.csv and press Enter. For example:

   NN-ESD# spl load acli config-csv esd-config.csv

   You can specify any filename using alpha-numeric characters. It is recommended that you save the file as a .csv file.

   The NN-ESD exports the editing configuration to the CSV file you specified, and stores it on the volatile file system (/ramdrv/ for 4500), (/var/ for NN-ESD).

5. Using FTP or SFTP, transfer the CSV file to your PC for viewing or editing.

6. Open the file using an application that supports a CSV file (for example, MS-Excel).
7. View or edit the CSV file as applicable. If you make changes, save the file, and import the CSV file back into the NN-ESD using the procedure *Importing a NN-ESD Configuration from a CSV File*.

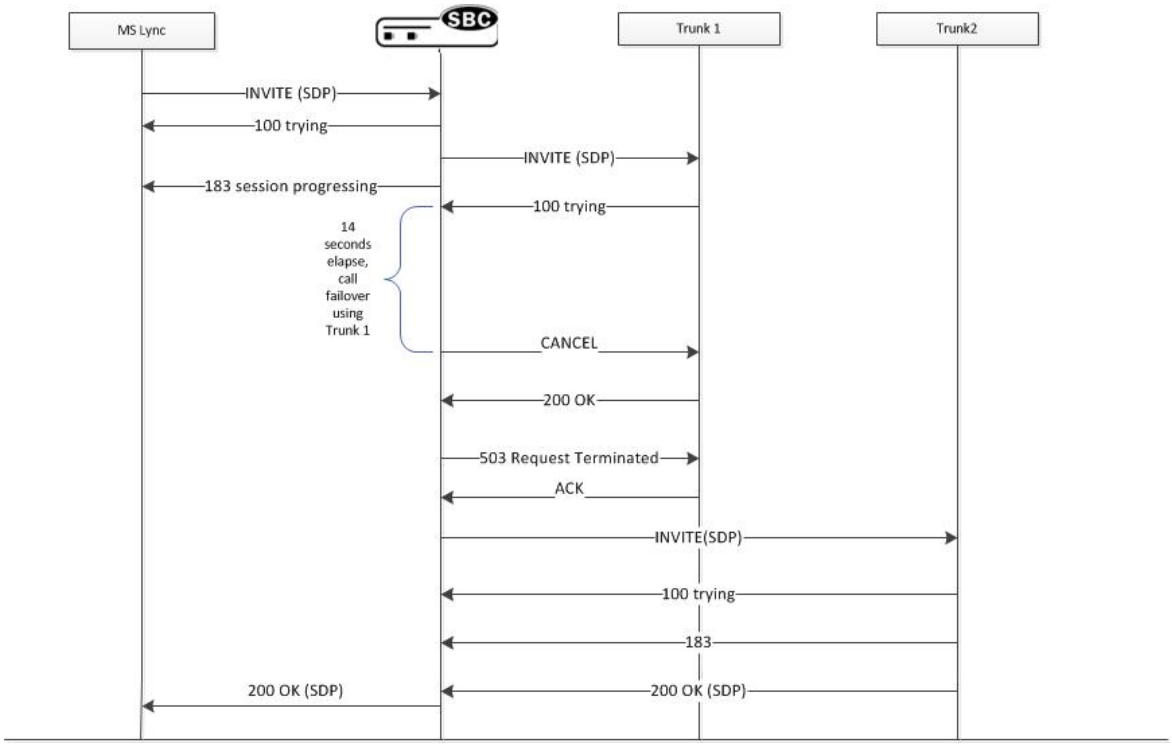
Lync Emergency Call SPL Plug-in

When using the Oracle Enterprise Session Border Controller (E-SBC) as a gateway for E911, service providers may require a Post-dial-delay of 6 seconds or greater to receive an 18x message and progress the session. Microsoft Lync emergency calls have an internal timer of 10 seconds to route advance to the alternate gateway in the event no 18x message is received. The Lync Emergency Call SPL plug-in responds to the initial INVITE with the 183 message, allowing theE-SBC to ensure normal call delivery when there is Post-dial-delay on egress routes that exceed the Lync emergency call timer.

When enabled, the return_183_on_initial_invite SPL plug-in sends a provisional 183 session progressing message to Lync when the E-SBC receives the initial INVITE request. This satisfies the 10 second emergency call timer.

The E-SBC monitors the primary and secondary trunks with a SIP OPTIONS ping. If the primary trunk is unavailable, the system automatically fails over to the second destination in session-group and completes the call.

In the following example, the E-SBC replies to an emergency call INVITE from Lync with a SIP 183 message. Lync moves the call and dialog to RFC 3261 Timer C (180 seconds), allowing sufficient time to complete the call and to find the nearest PSAP (Public Safety Answering Point) with the primary trunk. If the primary trunk is unavailable (The system monitors the two trunks with a SIP OPTIONS ping), the system routes the emergency call by failing over to a secondary trunk to complete the call.



Set Lync Emergency Call Options on Realms, Session Agents, and SIP Interfaces

You can configure the return_183_on_initial_invite option on each session-agent, realm-config, or sip-interface that interacts with Microsoft Lync. This option is not recognized in the global spl-config and is required for SPL functionality.

1. In Superuser mode, type configure terminal, and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router, and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent, and press Enter.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Type spl-config , and press Enter.

```
ACMEPACKET(session-agent)# spl-config
ACMEPACKET(spl-config)#
```

5. Type spl-options +return_183_on_initial_invite.

```
ACMESYSTEM(spl-config)# spl-options +return_183_on_initial_invite
```

6. Type done to save the configuration.

Example Playback-on-Refer Configuration

The following code block shows an example of a playback-on-refer configuration:

```
session-router
    session-agent
        spl-config
            spl-options          +return_183_on_initial_invite
```

SIPREC Extension Data Enhancements SPL

The SIPREC Extension Data Enhancements SPL provides additional header information in the originating SIP messages metadata sent to the Interactive Session Recorder. With this SPL, you can introduce more options for recording policy decisions when using the SIPREC feature of the Net-Net Session Border Controller (SBC). The enhanced metadata also allows for the realm-id to be used as an indicator of the recording account. The SPL also provides configurable values that collect additional header information to store in the metadata.

When the SPL is configured, the SIPREC Extension Data Enhancements SPL is only triggered upon INVITE requests, and stores the additional header information in the metadata that is sent to the Net-Net Interactive Session Recorder (NN-ISR). Metadata is a XML MIME attachment that describes recording details to the Net-Net ISR.

By default, the Extension-Headers SPL option collects only the Request-URI in a received INVITE. You can store additional header information by configuring the SPL with additional attributes in the spl-options under the global spl-config.. The values must be in a comma separated list enclosed in double quotation marks. For example:

```
Extension-Headers="P-Asserted-Identity,Diversion"
```

This configuration of the Extension-Headers option adds the originating Request-URI along with all P-Asserted-Identity and Diversion-Headers into the participant section of the metadata.

You can configure the LRE-Identifier SPL option to add an identifier of the logical remote entity (LRE) that triggered the recording to the <apkt:realm> element of the extension metadata. When configured with a value added, the value appears in place of the identifier. When configured without a value, the identifier of the logical remote entity is used. For example, session-agent will be the hostname, realm-config will be the realm, and sip-interface will be the realm name.

Note: Both options are required for the SPL to function properly.

Sample Metadata

The sample below shows metadata with new extension data added by the SIPREC Extension Data Enhancements SPL (New metadata appears in bold):

```
<?xml version='1.0' encoding='UTF-8'?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
        <datamode>complete</datamode>
        <session id="BYiC7uSZQGN3VQdzWI1HWw==">
                <associate-time>2012-06-26T13:44:13</associate-time>
        </session>
        <participant id="hq18GJs3TtJdhjPsfPNV8A=="
session="BYiC7uSZQGN3VQdzWI1HWw==">
                <nameID aor="sip:sipp@192.168.10.1">
                        <name>sipp</name>
                </nameID>
                <send>aD50KX+LTvxNzASg+/GQTg==</send>
                <associate-time>2012-06-26T13:44:13</associate-time>
                <extensiondata xmlns:apkt="http://acmepacket.com/
siprec/extensiondata">
                        <apkt:callingParty>true</apkt:callingParty>
                        <apkt:request-uri>sip:service@192.168.101.13
:5060    </apkt:request-uri>
                        <apkt:in-realm>net192</apkt:in-realm>
                        <apkt:header label=P-Asserted-Identity>
                                <value>sip:mike@acme.com</value>
                                <value>sip:bob@cisco.com</value>
                        </apkt:header>
                        <apkt:header label=Diversion>
<value>&lt;sip:jojo@divert.com&gt
;;happy=days;green=envy</value>
                                <value>&lt;sip:bebe@MediaTen.net&gt
;;green=monster;go=carts</value>
                                <value>&lt;tel:+8675309;night=mare&gt
;;gear=head;green=monitor</value>
                        </apkt:header>
                </extensiondata>
        </participant>
        <participant id="Ki6WEUi4TPRUPLtEaEhA7Q==" session=
"BYiC7uSZQGN3VQdzWI1HWw==">
                <nameID aor="sip:service@192.168.101.13">
                        <name>sut</name>
                </nameID>
                <send>f9NDVhyMTul+ePlM2SceQA==</send>
                <associate-time>2012-06-26T13:44:13</associate-time>
                <extensiondata xmlns:apkt="http://acmepacket.com/
siprec/extensiondata">
                        <apkt:callingParty>false</apkt:callingParty>
                </extensiondata>
        </participant>
        <stream id="aD50KX+LTvxNzASg+/GQTg=="
session="BYiC7uSZQGN3VQdzWI1HWw==">
                <label>65804</label>
                <mode>separate</mode>
                <associate-time>2012-06-26T13:44:13</associate-time>
        </stream>
        <stream id="f9NDVhyMTul+ePlM2SceQA=="
session="BYiC7uSZQGN3VQdzWI1HWw==">
                <label>65805</label>
                <mode>separate</mode>
                <associate-time>2012-06-26T13:44:13</associate-time>
        </stream>
</recording>
```

Setting SIPREC Extension Data Enhancement Options

The Extension-Headers option must be configured at the global level under spl-config. It is not recognized in the session-agent, realm-config, or sip-interface. If you update the list of headers to store in the Extension-Data SPL

option, you must perform a save and activate in order for the new option to take effect. This option is required for SPL functionality.

To configure the Extension-Headers option:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type spl-config and press Enter.

```
ACMEPACKET(system)# spl-config
ACMEPACKET(spl-config)#
```

4. Type spl-options +Extension-Headers="<value>" where <value> is the additional header information to store and press Enter. The default behavior stores only the Request-URI and realm-id.

```
ACMESYSTEM(spl-config)# spl-options +Extension-Headers="P-Asserted-
Identity,Diversion"
```

5. Type done to save your work.

Example Configuration

The following is an example of a SIPREC Extension Data Enhancement SPL configuration:

```
system
    spl-config
        spl-options          +Extension-Headers="P-Asserted                -
Identity,Diversion"
```

The LRE-Identifier option may be configured on each session-agent, realm-config, or sip-interface that interacts with the session recording server. This option is not recognized in the global spl-config. This option is required for SPL functionality.

To configure the LRE-Identifier option:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type sip-interface and press Enter.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Type spl-config and press Enter.

```
ACMEPACKET(sip-interface)# spl-config
ACMEPACKET(spl-config)#
```

5. Type spl-options +LRE-Identifier="<value>" where <value> is the additional header information to store and press Enter. The default behavior stores the identifier of the logical remote identity.

```
ACMESYSTEM(spl-config)# spl-options +LRE-Identifier
```

6. Type done to save your work.

Example Configuration

The following is an example of an LRE-Identifier option configuration:

```
session-router
    sip-interface
```

```
        spl-config
            spl-options          +LRE-Identifier
```

Universal Call Identifier SPL

The Universal Call Identifier SPL generates or preserves a UCID based on configuration. Once a UCID is generated or preserved, the system adds the value to all subsequent egress SIP requests within the session. You can also set the SPL to remove unwanted UCID headers to avoid duplicity in egress SIP requests.

Using the Universal Call Identifier SPL, you can identify requests within a particular session by manipulating the following vendor specific UCID headers:

- User-to-User
- Cisco-GUID
- Cisco-GUCID

The UCID is added as extension data to the session element of the recording's metadata when using SIPREC.

You must configure one of the following SPL options for it to be enabled:

- UCID-App-ID
- GUID-Node-ID
- GUCID-Node-ID

Each SPL option allows you to set an identifying value, as defined by the vendors. The SPL does not validate any input for the SPL options. It is the responsibility of the Administrator to set the correct value.

You may further modify the action of the SPL by adding replace-ucid or convert-to to your SPL options.

Note: The replace-ucid and convert-to options have no effect unless you also configure UCID-App-ID, GUID-Node-ID, or GUCID-Node-ID.

UCID-App-ID

The UCID-App-ID SPL option allows the Net-Net ESD to examine ingress SIP requests for the "User-to-User" header. When present, the header is transparently passed through the egress SIP message. If set to replace-ucid or the header is not present, the system generates a new value for "User-to-User".

You must set the value to a 2-byte hex-ascii value that represents the app ID. All input is truncated to 4 characters. Any characters outside the range of 0-9 and A-F will result in an invalid User-to-User header.

GUCID-Node-ID

The GUCID-Node-ID SPL option allows the Net-Net ESD to examine ingress SIP requests for the Cisco-GUCID header. When present, the header is transparently passed through the egress SIP message. If set to replace-ucid or the header is not present, the system generates a new value for Cisco-GUCID.

You must set the value to a 48-bit node ID in the version 1 UUID defined by RFC 4122. You can enter the value in decimal or hexadecimal notation. The value must be prefixed with 0x when hexadecimal.

GUID-Node-ID

The GUID-Node-ID SPL option allows the Net-Net ESD to examine ingress SIP requests for the Cisco-GUID header. If present, the header is transparently passed through the egress SIP message. The system generates a new value for Cisco-GUID if not present or the SPL option is set to replace-ucid.

You must set the value to a 48-bit node ID in the version 1 GUID defined by RFC 4122. You can enter the value in decimal or hexadecimal notation. The value must be prefixed with 0x when hexadecimal.

convert-to

The convert-to SPL option allows the Net-Net ESD to examine ingress SIP requests for multiple UCID headers. This option has no effect unless appended to another SPL option.

You must set the convert-to SPL option to one of the following values:

- Avaya—Removes all Cisco-GUCID and Cisco-GUID headers from egress SIP requests.
- GUID—Removes all User-to-User and Cisco-GUCID headers from egress SIP requests.
- GUCID—Removes all User-to-User and Cisco-GUID headers from egress SIP requests.

Example SPL Options

The following are examples of the Universal Call Identifier SPL.

Example 1

```
UCID-App-ID=0023,replace-ucid,convert-to=Avaya
```

This creates a User-to-User header based on a node ID of 23. Any value on the ingress side is replaced with the newly generated value. Removes all Cisco-GUID and Cisco-GUCID headers from egress messages.

Example 2

```
GUID-Node-ID=0x124578,convert-to=Guid
```

This creates a Cisco-GUID header if one does not exist in the ingress request. Removes all User-to-User and Cisco-GUCID headers from egress messages.

Sample Metadata

The following sample shows metadata with new extension data added by the Universal Call Identifier SPL:

```
<extensiondata xmlns:apkt=http:/acmepacket.com/siprec/extensiondata>
        <apkt:ucid>00FA08001900014E3E7D5C;encoding=hex</apkt:ucid>
</extensiondata>
<extensiondata xmlns:apkt=http:/acmepacket.com/siprec/extensiondata>
        <apkt:ucid>C0934BE72BF711D6800285D16359919A</apkt:ucid>
</extensiondata>
```

Configuring Universal Call Identifier Options

The SPL options must be configured on the ingress session-agent, realm-config, or sip-interface. If you update the values of the SPL options for the Universal Call Identifier SPL, you must perform a save and activate in order for the new option to take effect.

To configure the SPL options:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type session-router and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```

3. Type session-agent and press Enter.

   ```
   ACMEPACKET(session-router)# session-agent
   ACMEPACKET(session-agent)#
   ```

4. Type spl-config and press Enter.

   ```
   ACMEPACKET(session-agent)# spl-config
   ACMEPACKET(spl-config)#
   ```

5. Type spl-options +UCID-App-ID=”<value>” where <value> is the additional header information to store and press Enter. The default behavior stores only the Request-URI and realm-id.

```
ACMESYSTEM(spl-config)# spl-options +UCID-App-ID=0023
```

6. Type done to save your work.

Example Configuration

The following is an example of an Universal Call Identifier option configuration:

```
session-router
    session-agent
        spl-config
            spl-options          +UCID-App-ID=0023
```

Comfort Noise (CN) Generation SPL

Comfort noise (CN) is the noise in a Real Time Transport Protocol (RTP) message (defined in RFC 3389) that is played to prevent a user from hearing completely dead silence on the connection. The Session Description Protocol (SDP) negotiates this RTP message containing the comfort noise using payload type 13 and an rtpname of "CN".

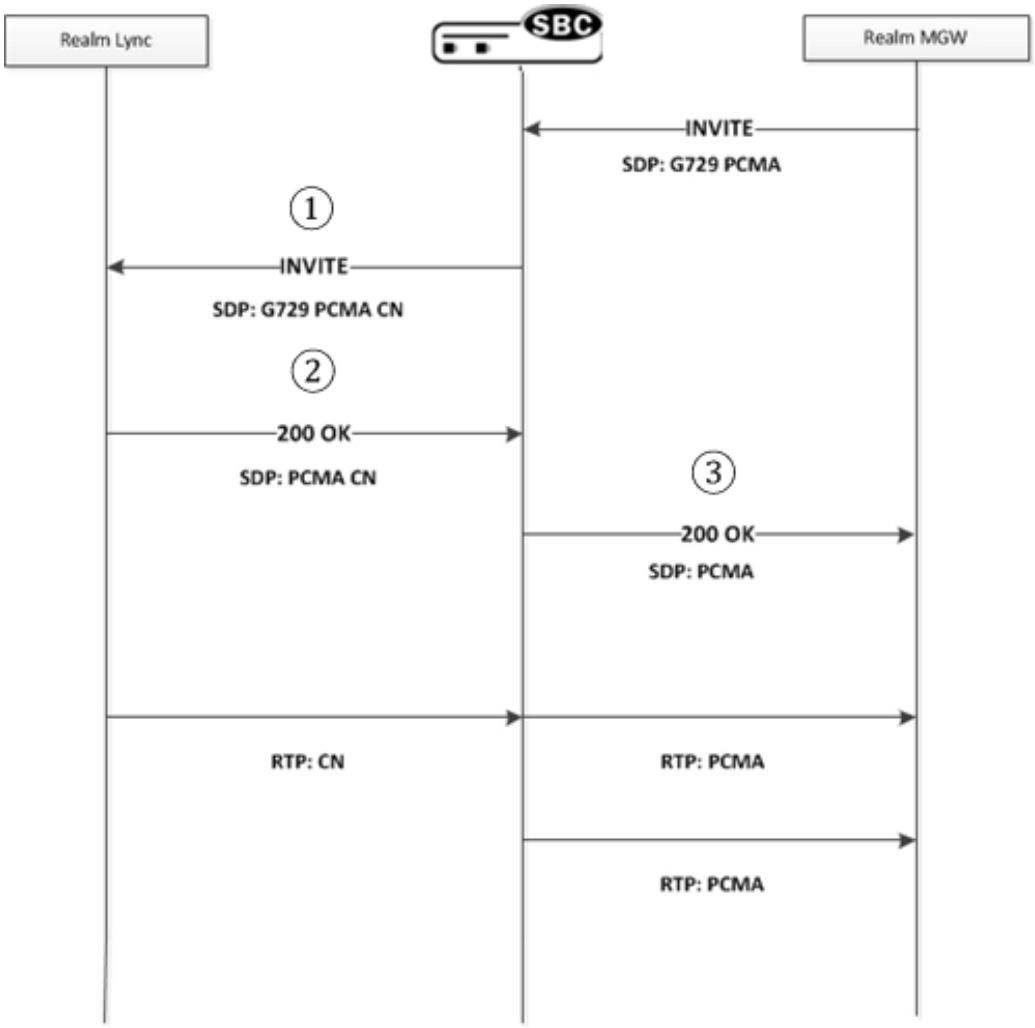
However, when CN is received, normal RTP ceases. Thus, with no RTP traffic, guard timers may trigger and cause the call to be terminated. To correct this, you can load a Comfort Noise Generation SPL that allows the Net-Net ESD to generate "noise" RTP using the normal audio codec when it receives a CN indication.

The CN Generation SPL must be loaded manually according to the procedures described in *Loading and Enabling the SPL*. After loading the SPL on the Net-Net ESD, comfort noise is added and removed from the SDP to allow for proper negotiation. If properly negotiated in SDP, CN interworking facility (IWF) is enabled in the media flows allowing the Net-Net ESD to generate noise RTP when CN is received.

Note: CN generation configuration is supported on realms only.

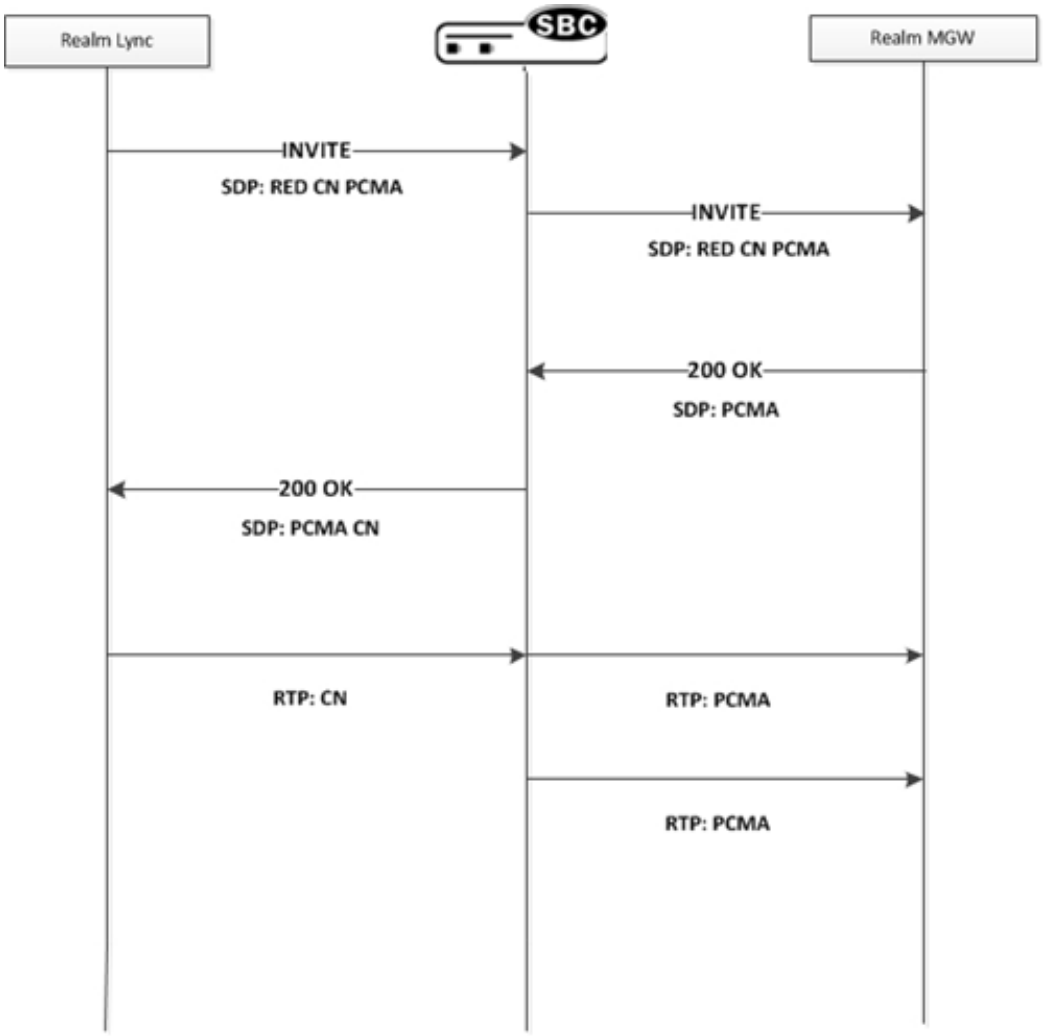
The following describes two different cases of how the Net-Net ESD performs the SDP manipulation using the CN Generation SPL.

Case 1



Process	Description
①	SDP offer received from the realm that has comfort-noise-generate enabled. If the SDP offer contains CN, no IWF is required. If it does not contain CN, and at least one of the offered audio codecs is PCMU or PCMA, CN is added in outgoing SDP offer.
②	If SDP Answer contains the CN codec and topmost audio codec is PCMU or PCMA, Net-Net ESD enables CN IWF.
③	Net-Net ESD strips CN from outgoing SDP Answer.

Case 2



Process	Description
①	SDP offer is sent to a realm that has comfort-noise-generate enabled. If CN was not offered, IWF cannot be performed. If CN is in the offer, the Net-Net ESD forwards the offer to the outbound side.
②	SDP Answer is received from a realm that has comfort-noise-generate enabled. If it contains CN, no IWF is required because both sides support CN. If there is no CN in the Answer, and the topmost audio codec is PCMU or PCMA, the Net-Net ESD enables CN IWF.
③	If CN IWF is enabled, the Net-Net ESD adds CN to the outgoing SDP Answer.

Configuring the CN Generation SPL

Use the following procedure to configure the CN Generation SPL on the Net-Net ESD.

- The CN Generation SPL can be installed on the Server Edition and VMWare (preferred hypervisor) platforms, only.
- You must have the SPL loaded and enabled using the procedure in *Loading and Enabling the SPL*.

To configure the CN Generation SPL:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type realm-config and press Enter.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Type spl-options +comfort-noise-generate and press Enter.

```
ACMESYSTEM(realm-config)# spl-options +comfort-noise-generate
```

5. Type done to save your work.

Example Configuration

The following is an example of a CN Generation SPL configuration.

```
media-manager
    realm-config
        identifier          SP
        addr-prefix         172.16.0.0/16
        network-interfaces      Core1:0
        mm-in-realm     enabled
        spl-options     comfort-noise-generate
```

High Availability (HA) Support

High Availability (HA) supports the use of Comfort noise. The codec encoding (PCMU/PCMA), codec ptime, and CN generation enabled/disabled state is exchanged with the standby in Middlebox Control Daemon (MBCD). If CN generation occurs in a call flow, and a switchover occurs, the CN generation stops until the next CN message is received.

Licensing Information

The following licensing applies to the CN Generation SPL.

Process	Description
Package name	G711
License category	Open Source
Package version	N/A
Vendor	Sun Microsystems
Applicable license	This source code is a product of Sun Microsystems, Inc. and is provided for unrestricted use. Users may copy or modify this source code without charge. SUN SOURCE CODE IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. Sun source code is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modification or enhancement. SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY THIS SOFTWARE OR ANY PART THEREOF. In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even if Sun has been advised of the possibility of such damages.

Process	Description
	Sun Microsystems, Inc. 2550 Garcia Avenue Mountain View, California 94043
Software builds	SC[z] BC[z]640
Purpose	Conversion from linear samples to alaw/ulaw
Used in modified or unmodified form	Unmodified
Used internally or distributed	Distributed with Product Binaries
Location in source code tree	linux/kernel/modules/acme
Used by itself or in combination with other software	Used exclusively and tightly integrated into acme.ko. The code is compiled as part of acme.ko.
Link to website hosting Software	N/A
License requires notice provided in product documentation?	No

Maintenance and Troubleshooting Commands for SPLs

This section provides information about how to troubleshoot and collect information about your SPL.

show SPL

The ACLI show spl command displays:

- The version of the SPL engine.
- The filenames and version of the SPLs currently loaded on the Net-Net ESD.
- The signature state of each SPL
- The system tasks for which each loaded SPL interacts, enclosed in brackets.

```
ACMEPACKET# show spl
SPL Version: C2.0.0
[sipd] File: LyncEmergencyCall.1.0.spl version: 1.0 signature: signed and
valid
ACMEPACKET# show spl sipd
SPL Version: C2.0.0
[sipd] File: LyncEmergencyCall.1.0.spl version: 1.0 signature: signed and
valid
```

SPL Signature State

All SPLs must be signed by Acme Packet for authenticity.

show running-config spl-config

The ACLI show running-config spl-config displays SPL specific configuration information on the system.

```
ACMEPACKET# show running-config spl-config
spl-config
        spl-options
        plugins
                name                            LyncEmergencyCall.1.0.spl
        last-modified-by                admin@216.41.24.2
        last-modified-date              2012-10-12 15:31:05
```

show directory code spl

The ACLI show /code/spl command displays the SPLs stored in the /code/spl directory.

```
ACMEPACKET# show directory /code/spl
Listing Directory /code/spl:
drwxrwxrwx  1 0       0                4096 Aug 13 10:07 ./
drwxrwxrwx  1 0       0                4096 Aug 19 22:25 ../
-rwxrwxrwx  1 0       0                3163 Aug 13 10:07 LyncEmergencyCall.
1.0.spl
```

show spl-options

The ACLI show spl-options command displays SPL-specific options registered by an SPL.

```
ACMEPACKET# show spl-options
   1. return_183_initial_invite: Returns a 183 provisional response when a
emergency call is placed through Lync [LyncEmergencyCall.1.0.spl,config]
```

Deleting SPLs

Deleting files from /code/spl must be performed via FTP/SFTP; there is no means to delete files from the ACLI.

SPL Log Types

SPL log messages can often be found in the log file for the system task to which the SPL applies when that task is set to DEBUG level. You can find the output specific to SPL by the identifying prefix [SPL].

```
Aug 30 15:06:07.454 [SPC] Executing SPL callback from file:
SipHeaderExtensionMetadata.1.2.spl
Aug 30 15:06:07.454 [SPL] Checking for LRE-Identifier to match triggered
session-recording-server
Aug 30 15:06:07.454 [SPC] Creating table of name
'AcmeSipServerTransDataTable' with key [0x34522878]
Aug 30 15:06:07.454 [SPC] Creating new temporary session table of key
[_SESSION_0x34522878]
Aug 30 15:06:07.454 [SPL] SIP Interface ingressSIP has option
Aug 30 15:06:07.454 [SPL] Storing data from message to insert into metadata
```

Emergency Location Identification Number (ELIN) Gateway Support

An ELIN-capable gateway supports connection to a qualified E911 service provider. The connection supports PSTN-based E911 functions, including user callback when there is a disconnect. Enterprises often deploy ELIN numbers based on physical location to locate the physical source of a 911 call. By using multiple ELINs, an enterprise can support multiple, simultaneous E911 calls.

Typically, an enterprise purchases multiple ELIN numbers. An ELIN gateway replaces VoIP extension URIs with ELIN numbers and maintains the mapping. For example, if an emergency service replied to a VoIP URI without using an ELIN gateway, the reply would be delayed or fail. An ELIN gateway can use its mapping to translate the ELIN

number back to the VoIP extension from within the enterprise session network. The gateway can immediately forward the call back to the original client.

The Oracle Enterprise Session Border Controller supports E911 ELIN for Lync-enabled Enterprises using the ELIN_Gateway SPL option. Enable this option in the global SPL configuration. The Oracle Enterprise Session Border Controller supports up to 300 ELIN numbers simultaneously and it can reuse numbers allowing a greater number of emergency calls.

How the Emergency Location Identification Number (ELIN) SPL Works

When a Lync client places a 911 emergency call through a mediation server to a Oracle Enterprise Session Border Controller, the server indicates the emergency status in the priority field and provides a list of ELIN numbers. When the ELIN gateway module is enabled, the Oracle Enterprise Session Border Controller intelligently selects a particular ELIN number and uses it as the ANI in the “From” field SIP URI in the outgoing INVITE.

The Oracle Enterprise Session Border Controller preserves the mapping of used ELIN numbers in an internal table. This table includes the ELIN number, the caller (VoIP extension), the “in-use” count, and a timer field. The Oracle Enterprise Session Border Controller retains these mappings for a configurable time period ranging from 30 to 60 minutes after the call is terminated. The default is 30 minutes. When the timer expires, the entry is purged from the table. The timer field shows the time of day that the timer started.

You can view the current ELIN table at any time using the ACLI command spl show sip elins.

After the Lync client call is disconnected, the 911 service may call back using the number provided in the “From” field of the original INVITE. This presence of this number in its ELIN number table allows the Oracle Enterprise Session Border Controller to route the call back to the original caller.

Number Reuse

The Oracle Enterprise Session Border Controller can use an ELIN number for multiple calls. When a call that requires an ELIN mapping arrives at the Oracle Enterprise Session Border Controller, it checks to see if the numbers presented by the mediation server are in use. If a number is not in use, it simply uses that number. A number is not in use if it is not in the table or its “used count” is 0. An entry’s used count is zero when its not in use and its purge timer has not yet expired.

If all numbers are in use, the Oracle Enterprise Session Border Controller employs a means of reusing a number, incrementing its used count for each additional call. The selection process proceeds in the following order:

1. If the “caller” is in the ELIN table, the Oracle Enterprise Session Border Controller selects that mapping.
2. The Oracle Enterprise Session Border Controller selects the number with the lowest “ELIN count”.

If an ELIN number is used by multiple calls, it maps callback attempts to that ELIN number to the client that was last associated with the number.

Error Handling

Lync mediation servers always expect 503 “Service Unavailable” as an error message to a failed ELIN call. There is a variety of error messages that the network may send back when a call fails. For the purposes of Lync support, the Net-Net ESD sends 503 “Service Unavailable” to indicate call failure to a mediation server, regardless of the error it receives.

Configure the Emergency Location Identification Number (ELIN) Gateway Option

The ELIN-Gateway option must be configured at the global level under spl-config or by way of the Web GUI. The ELIN-Gateway option is not recognized in the session-agent, realm-config, or sip-interface.

Determine the preferred length of time to retain ELIN mappings within the Oracle Enterprise Session Border Controller. The range is from 30 to 60 minutes. The default is 30 minutes.

To configure the ELIN Gateway option:

1. In Superuser mode, type configure terminal and press <Enter>.
   ACMEPACKET# configure terminal
2. Type system, and press <Enter>.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type spl-config, and press <Enter>.

```
ACMEPACKET(configure)# spl-config
ACMEPACKET(spl-config)#
```

4. Type spl-options +Extension-Headers="<value>" where <value> is the additional header information to store, and press <Enter>. The default behavior stores only the Request-URI and realm-id.

```
ACMEPACKET(spl-config)#spl-options +Elin-Gateway=60
```

5. Type done to save your work.

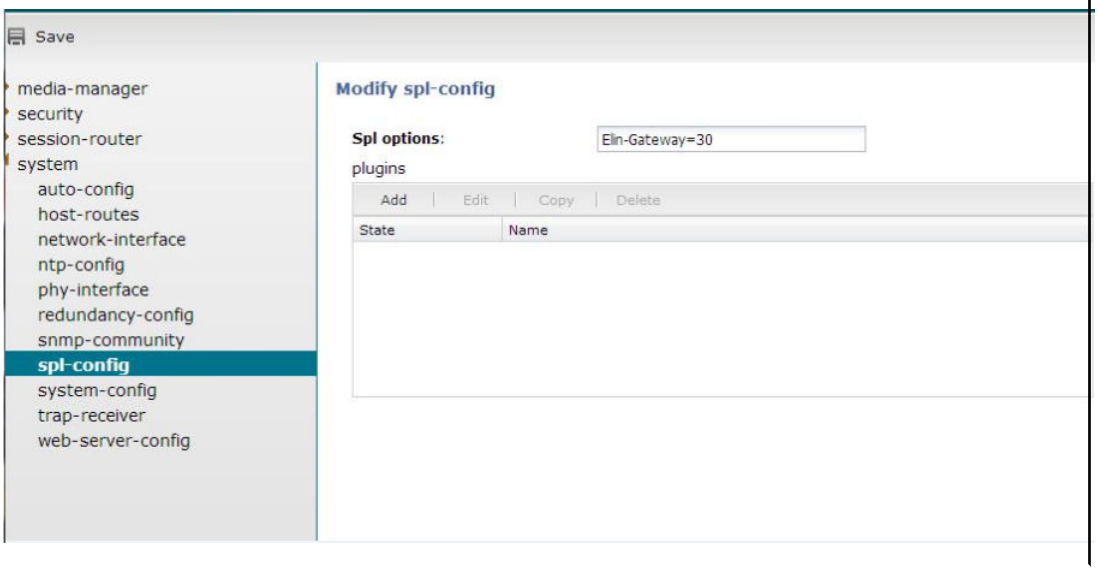
The following is an example of an Elin_Gateway SPL configuration:

```
system
    spl-config
        spl-options           Elin-Gateway=60
```

The following is an example of the ACLI command spl show sip elins.

```
ACMEPACKET#show sip elins
Elin:1111442231
Count:0 From:5555221134 Time:1380490337.8292
--------------
Elin:2222882232
Count:0 From:6666111234 Time:1380490770.4083
```

To configure the ELIN-Gateway option using the Web GUI, select spl-config, add a config, and save.



Avaya Session Manager (SM) Redundancy

To support redundancy in Avaya SM deployments, the Oracle Enterprise Session Border Controller can use the mechanisms for maintaining multiple connections defined in RFC 5626. In an Avaya SM deployment, this scenario is referred to as Dual Registration.

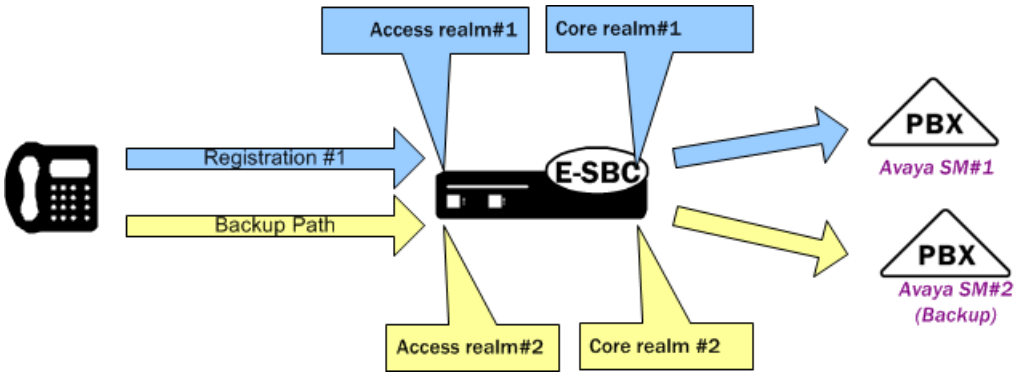
RFC 5626 specifies a method of maintaining connections between UAs and proxies, and outlines a general means for UAs to establish connection redundancy. The Oracle Enterprise Session Border Controller can use RFC 5626 specifically for redundancy in Avaya SM Dual Registration deployments. Such a deployment allows the network to continue to provide service by way of a redundant Avaya SM, when the primary Avaya SM stops responding.

Oracle Enterprise Session Border Controller configuration requires adding the rfc 5626 SPL option. In addition to adding the SPL option, the Oracle Enterprise Session Border Controller configuration design separates Avaya SMs and UA traffic by way of using realms.

How Avaya Session Manager (SM) Redundancy Works

To support Avaya SM redundancy, you configure multiple realms on the access side and the core side of the Oracle Enterprise Session Border Controller. These realms create the primary path and backup path for accessing a redundant Avaya SM.

Consider two Avaya SMs deployed for redundancy. You configure a core side realm for each Avaya SM and you configure two access side realms. Each access side realm is associated with the applicable core side realm, to which a UA sends registration messaging. The following illustration shows this configuration.



The operational scenario consists of the Avaya SM infrastructure providing configuration information to the UAs. The information includes the 2 proxy addresses, targeting the Oracle Enterprise Session Border Controller access-side interfaces. The UA knows which proxy is for the primary path, and sends initial REGISTER messages by way of that path. While the primary Avaya SM is up, the UA manages all registration exchanges, including refresh and re-register procedures, on the primary path. If the primary Avaya SM stops responding, the infrastructure informs the UA that it needs to register with the backup Avaya SM. The UA registers with the backup Avaya SM using the backup path.

The UA, by way of configuration, populates the backup registration and subsequent registration messages so that the Avaya SM infrastructure knows that the registrations are for the same UA. Key elements of the messaging and their use by the Avaya dual registration infrastructure include:

- reg-id - A contact header field parameter value that specifies individual registrations. UAs use unique reg-id values to specify registrations for individual flows.
- instance-id (+sip.instance) - A URN within the contact header that specifies a UA instance. UAs use the same Instance ID information in REGISTER exchanges to indicate that the registrations belong to the same UA.
- Route - The target proxy for the message. The UA uses route headers to define the separate paths to the Oracle Enterprise Session Border Controller.

The Avaya SM uses reg-id in conjunction with instance-ID to manage dual registrations. By keeping instance-ID the same and sending a new reg-id, the infrastructure recognizes that a redundant registration was generated because a session manager switchover occurred.

Normally, multiple reg-ids based on a single contact would trigger a "move" procedure. The presence of a single instance-ID tells the infrastructure that the reg-id change does not indicate a move.

The following example REGISTERs depict the population of these elements for the purposes of an Avaya dual registration scenario.

```
 REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.2;branch=z9hG4bK-bad0ce-11-1036
```

```
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=d879h76
To: Bob <sip:bob@example.com>
Call-ID: 8921348ju72je840.204
Supported: path, outbound
Route: <sip:ep1.example.com;lr>
CSeq: 1 REGISTER Supported: path, outbound
Contact: <sip:line1@192.0.2.2;transport=tcp>; reg-id=1;
;+sip.instance="<urn:uuid:00000000-0000-1000-8000-000A95A0E128>" Content-
Length: 0
```

Note the following redundant registration. The registration includes a different route header for the second Oracle Enterprise Session Border Controller realm. It also includes a new reg-id and the same instance-ID.

```
 REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.2;branch=z9hG4bK-bad0ce-11-1036
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=d879h76
To: Bob <sip:bob@example.com>
Call-ID: 8921348ju72je840.204
Supported: path, outbound
Route: <sip:ep2.example.com;lr>
CSeq: 1 REGISTER Supported: path, outbound
Contact: <sip:line1@192.0.2.2;transport=tcp>; reg-id=2;
;+sip.instance="<urn:uuid:00000000-0000-1000-8000-000A95A0E128>" Content-
Length: 0
```

Registration Caching

Enabling the RFC 5626 SPL option causes the Oracle Enterprise Session Border Controller to store a single, entire contact header in its registration cache for the AOR. When an Avaya SM switchover occurs, the Oracle Enterprise Session Border Controller updates the AOR by replacing the contact header with the new one. The Oracle Enterprise Session Border Controller does not store more than one contact per AOR. The Oracle Enterprise Session Border Controller establishes a flow with only the active Avaya SM.

Session Manager Mapping

The Oracle Enterprise Session Border Controller (SBC) supports mapping between multiple session managers and multiple SBCs. Such mapping allows the SBC to work in a redundant network configuration where you can map:

- The primary session manager to the primary SBC IP address
- One or more redundant session managers to one or more redundant SBCs

To map a redundant session manager to a redundant SBC, map the private IP address of the redundant session manager to the public SIP IP address configured in HTTP-ALG > Public on the SBC. For instructions, see "Map a Session Manager to a Session Border Controller."

Map a Session Manager to a Session Border Controller

You can map one or more session managers to an Oracle Enterprise Session Border Controller (SBC) to provide redundancy and load balancing. Map the private IP address of the session manager to the public SIP interface IP address of the SBC.

Before You Begin

- Note the private IP address of the session manager and the public SIP interface IP address of the session border controller that you want to map.

Procedure

1. From the command line, type configure terminal, and press ENTER.
2. Type session-router, and press ENTER.

3. Type http-alg, and press ENTER.
   The system displays a numbered list of configured HTTP-Application Layer Gateway (ALG) objects.
4. Type the number of the HTTP-ALG object that you want to edit.
   The system displays the configuration values for the selected object.
5. Type session-manager-mapping, and press ENTER.
   The system displays a numbered list of configured HTTP-Application Layer Gateway (ALG) objects.
6. Type session-manager <IP address>, and press ENTER.
7. Type public-interface. and press ENTER.
8. Type ip-address <SBC public SIP IP address>, and press ENTER.
9. Type sip-port <port for SIP calls>, and press ENTER.
10. Type sip-transport-protocol <UDP, TCP, TLS>, and press ENTER.
11. Type done, and press ENTER.
12. Type exit, and press ENTER.
13. Type done, and press ENTER.
14. Type exit, and press ENTER.
15. Type done, and press ENTER.
16. Type show http-public-interface, and press ENTER
    The system displays the public interface values.
17. Type Done, and press ENTER to save the public interface values.
18. Exit, Save, and Activate the configuration.

Configure Avaya Session Manager (SM) Redundancy

The rfc5636 SPL option allows the Oracle Enterprise Session Border Controller to support Avaya Dual Registration for establishing redundant UA registration.

The rfc5626 option must be configured at the global level under spl-config or using the Web GUI. The rfc5626 option is not recognized in the session-agent, realm-config, or sip-interface.

To configure the rfc5626 option:

1. In Superuser mode, type configure terminal and press <Enter>.
   ACMEPACKET# configure terminal
2. Type system and press <Enter>.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type system and press <Enter>.

```
ACMEPACKET(system)# spl-config
ACMEPACKET(spl-config)#
```

4. Type rfc5626 and press <Enter>.

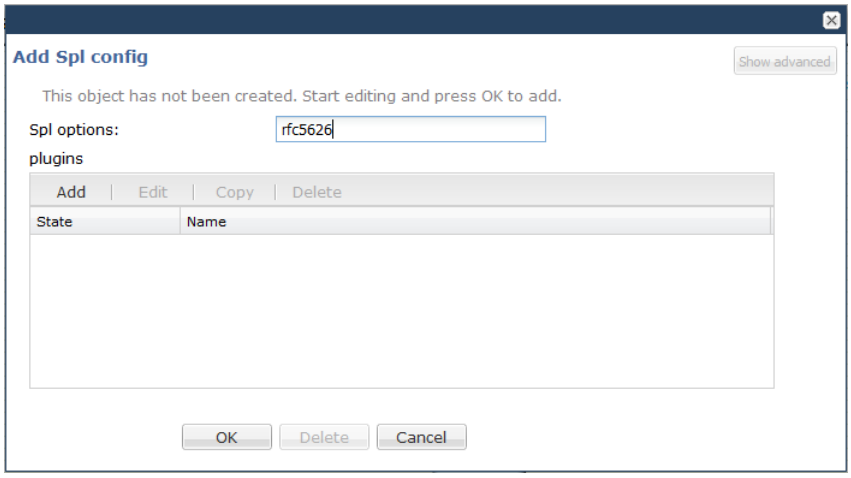
```
ACMEPACKET(spl-config)# rfc5626
ACMEPACKET(spl-config)#
```

5. Type done to save your work.
6. Save and Activate your configuration.

The following example shows an rfc5626 SPL configuration:

```
system
    spl-config
        spl-options          rfc5626
```

You can also configure the rfc5626 option using the Web GUI. The procedure consists of simply opening the spl-config dialog, adding the SPL option, then saving and activating.


Add Spl config
Show advanced
This object has not been created. Start editing and press OK to add.
Spl options:
rfc5626
plugins
Add
Edit
Copy
Delete
State
Name
OK
Delete
Cancel

A

Additional SNMP Support

Overview

The SNMP agent is part of the Net-Net ESD image and runs as a thread in the application. The SNMP support includes SNMP Get/Set and Trap operations in either v1v2 mode or secure trap mode.

- v1v2 mode. The SNMP agent supports SNMPv1 and v2c queries. SNMP traps are transmitted in v2c format only.
- secure trap mode. The SNMP agent supports only SNMP traps transmitted in v3 format with mandatory authentication and privacy.

MIB Changes

The following changes were made to existing Acme Packet MIBs for this Net-Net ESD release.

- ap-products.mib: a product series added called apNetNetOSSeries. For the Phase 1 release, a platform was defined called apNetNetOS.
- ap-license.mib: is not supported in current release.
- ap-entity-vendortype.mib: is not supported in Net-Net ESD because it is a software-only product.
- ap-env-monitor.mib: is not supported in Net-Net ESD because it is a software-only product.

SNMPv3 Support

The Oracle Enterprise Session Border Controller (E-SBC) supports SNMPv3, which provides the SNMP agent and SNMP Network Management System (NMS) with authentication, privacy, and access control during the delivery of secured traps. Currently, SNMPv3 traps are supported on the Net-Net ESD; SNMPv3 Get/Get-Bulk/Set actions are not supported at this time.

By default, the E-SBC supports SNMPv1v2. If you want to retain the existing SNMPv1v2 behavior, you do not need to update configuration. You can enable SNMPv3 at any time, at which point SNMPv1v2 configurations are ignored, and only SNMPv3 encrypted traps are sent to the associated external SNMP managers. Snmp-agent-mode, an attribute under system-config, allows you to select the mode that you want.

Authentication and Privacy

SNMPv3 employs a User-Based Security Model (USM). The two protocols used for authentication and privacy are:

- Authentication—HMAC-SHA-96
- Privacy—CBC-DES

Four parameters generate keys under the designated algorithm:

- SNMPEngineID—The unique identifier for the SNMP Engine. This value is a specially formatted string for use in the SNMP.
- User name—The user's name as defined under snmp-user-entry.
- Authorization password—The authorization password configured under the snmp-user-entry configuration. This parameter is used to derive the authentication key.
- Privacy password—You set the privacy password in the snmp-user-entry configuration. It is used to derive the password key.

Password-to-Key Conversion

There are two distinct passwords in SNMPv3. The authentication password is manipulated using the HMAC-SHA-96 algorithm to produce a key used to authenticate the trap. Authentication ensures the identity of the user and that the trap has not been tampered with in transit. Likewise, the privacy password is manipulated using the CBC-DES algorithm to ensure message privacy.

One user is associated by a name, an authentication password and a privacy password. These three parameters are always consistent for the user and can be used across multiple SBCs. The key generation differs from one SBC to another due to the varying SNMPEngineIDs. This ensures that a compromised key for one SBC does not compromise the keys for other SBCs associated with the same user.

Enabling SNMPv3

The table below gives a brief overview of the SNMPv3 configuration on your Net-Net SBC. The Caveats column describes the SNMPv1v2 configuration attributes that are ignored if SECURE-TRAP mode is enabled.

Configuration	Description	Caveat
snmp-agent-mode	Set this attribute to SECURE-TRAP to enable SNMPv3.	Once SNMPv3 is enabled, the snmp-community and community-name attributes are ignored.
snmp-engine-id-suffix	Set this attribute as a string to customize and uniquely identify the SNMP Engine.	The show snmp-info command has been expanded to include the SNMP Engine Base, the SNMP Engine Suffix, and the SNMP Engine ID.
snmp-user-entry	Enter the user name, authorization password and privacy password.	The user, as defined in this object, must be added to the attribute user-list under trap-receiver in order to receive secured traps.
trap-receiver	Configure a trap-receiver with the IP address of the NMS that receives secured traps.	
user-list	Add users who are authorized to receive secured traps.	If instances of snmp-user-entry are configured, but no users are listed under user-list, a warning message is sent during a verify-config execution.

Retaining Existing SNMPv1v2 Behavior

If you are upgrading to software version C6.3.0 or above and want to retain your existing SNMP configurations, you do not need to take any action. The Net-Net SBC sets snmp-agent-mode to V1V2 by default, disabling all SNMPv3 configurations.

Downgrading Software After Enabling SNMPv3

If you enable SNMPv3 on an SBC running C6.3.0 or above, and you downgrade to a previous software version, the software does not recognize the SNMPv3 configuration objects or attributes.

Consideration for HA Nodes

Key pairs are generated based on the user and SNMPEngineID. In the event of a switchover, the SNMPEngineID will vary. The user's NMS should be updated with the SNMPEngineID of the standby SBC. No action is required on the Net-Net ESD.

Enabling SNMPv3

This section shows you how to enable SNMPv3 on your system, how to add users, and how to add users to authorized trap receivers.

To enable SNMPv3 by adding a user and passwords:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type snmp-user-entry and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# snmp-user-entry
   ACMEPACKET(snmp-user-entry)#
   ```

   From this point, you can configure the SNMP user entry parameters. To view all SNMP user entry parameters, enter a ? at the system prompt.

4. user-name—Enter the user name of a person who is authorized to receive secure traps. Valid values are alpha-numeric characters. Default is blank.
5. auth-password—Enter the authentication password associated with the user-name value. The Net-Net ESD uses this password to authenticate the user before receiving secure traps. Valid values are alpha-numeric characters. Default is blank.
6. priv-password—Enter the privacy password associated with the user-name value. The Net-Net ESD uses this password to keep the session private for the user that is receiving the secure traps. Valid values are alpha-numeric characters. Default is blank.

   The following is an example of an SNMP user entry configuration. Parameters not described in this section are omitted below.

   ```
   snmp-user-entry
   user-name    jsmith
   auth-password    ********
   priv-password    ********
   ```

Trap Receiver Configuration

To configure trap receivers:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type system and press Enter to access the system-level configuration elements.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type trap-receiver and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(system)# trap-receiver
   ACMEPACKET(trap-receiver)#
   ```

   From this point, you can set trap receivers.

   The following is an example of a trap receiver configuration. Parameters not described in this section are omitted below.

```
trap-receiver
        ip-address                      10.0.1.42:162
        filter-level                    All
        community-name                  public
    user-list    jsmith,carolM,gcather
```

4. ip-address—Set the IPv4 address of an authorized NMS. This parameter is the IPv4 address of an NMS where traps are sent. If you do not specify a port number, the default SNMP trap port of 162 will be used.
5. filter-level—Set the filter level threshold that indicates the severity level at which a trap to be sent to this particular trap receiver. The default for this parameter is critical.

   Example: A trap with a severity level of Critical is generated, the SNMP agent will only send this trap to NMSs that are configured in a trap-receiver element and have a filter-level parameter of Critical.

   The following table maps Syslog and SNMP alarms to trap receiver filter levels.

Filter Level	Syslog Severity Level	(SNMP) Alarm Severity Level
Critical	Emergency (1) Critical (2)	Emergency Critical
Major	Emergency (1) Critical (2) Major (3)	Emergency Critical Major
Minor	Emergency (1) Critical (2) Major (3) Minor (4)	Emergency Critical Major Minor
All	Emergency (1) Critical (2) Major (3) Minor (4) Warning (5) Notice (6) Info (7) Trace (8) Debug (9)	Emergency Critical Major Minor Warning

   When configuring the trap-receiver element for use with the Element Management System (EMS), Oracle recommends that you set the filter-level parameter to All for that configuration element that includes EMS servers.
6. community-name—Set the community name to which this trap receiver belongs. This community must be defined in the SNMP community element.
7. user-list—For SNMPv3, specify a list of users that have authorized permissions to receive secure traps. Enter the user names as comma-separated values. For example:

```
ACMEPACKET(trap-receiver)# user-list jsmith,carolm,glather
```

Note: If instances of snmp-user-entry are configured, but no users are listed under user-list, a warning message is sent during a verify-config execution.

Acme Packet Net-Net ESD MIB (ap-usbcsys.mib)

The following table describes the SNMP GET query names for the Acme Packet Net-Net ESD MIB (ap-usbscsys.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apUsbcSysMIBObjects (1.3.6.1.4.1.9148.3.17.1)		
Object Identifier Name: apUsbcSysObjects (1.3.6.1.4.1.9148.3.17.1.1)		
apUsbcSysCpuUtilAll	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.1.0	Percentage of CPU utilization.
apUsbcSysCpuCount	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.2.0	Number of CPUs for this system.
apUsbcSysCpuSpeedMHz	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.3.0	Speed in MHz of the CPUs for this system.
apUsbcSysMemSzMB	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.4.0	Number of megabytes of all CPUs for this system.
apUsbcSysMemSzGB	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.5.0	Number of gigabytes of all CPUs for this system.
apUsbcSysAppMemUtil	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.6.0	Percentage of total memory utilization by applications.
apUsbcSysKernelMemUtil	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.7.0	Percentage of total memory utilization by the kernel.
apUsbcSysMyBogoMips	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.8.0	Processor speed in mips(millions of instructions per seond). Speed is calculated by the kernel at boot time.
apUsbcSysAllBogoMips	apUsbcSysObjects: 1.3.6.1.4.1.9148.3.17.1.1.9.0	Sum of all bogo mips (millions of instrctuctions per second) of all CPUs for this system.
Object Identifier Name: apUsbcSysCpuTable (.1.3.6.1.4.1.9148.3.17.1.1.10.1)		
Object Identifier Name: apUsbcSysCpuEntry (.1.3.6.1.4.1.9148.3.17.1.1.10.1.1)		
apUsbcSysCpuNum	apUsbcSysCpuEntry: 1.3.6.1.4.1.9148.3.17.1.1.10.1.1.1	CPU number + 1 of this entry.
apUsbcSysCpuUtil	apUsbcSysCpuEntry: 1.3.6.1.4.1.9148.3.17.1.1.10.1.1.2	Percent of CPU utilization for this CPU.

B

RTC Support

This appendix summarizes real-time configuration (RTC) support status for the Oracle Enterprise Session Border Controller . The table below lists which configuration elements are supported by RTC and which are not.

ACLI Configuration Elements	Parameter Details
Access Control	
Accounting Config	
Authentication	
Certificate Record	
Class Profile	
Codec Policy	
DNS ALG Service	
DNS Config	
Enum	
External Policy Server	
H.323	The following H.323 stack subelement parameters are not RTC supported in that, if you save and activate a configuration, calls already in progress will be dropped. A reboot is required. • state • isgateway • realm-id • assoc-stack • options • proxy-mode • local-ip • max-calls • max-channels • registration-ttl • terminal-alias

ACLI Configuration Elements	Parameter Details
	• prefixes • ras-port • q931-port • auto-gk-discovery • multicast • gatekeeper • h245-tunneling • gk-identifier • alternate-transport • q931-max-calls • filename
Host Route	
IPSEC	
IWF	
Licensing	
Local Policy	
Local Response Map	
Local Routing Config	
Media Manager	The Media Manager element is supported with the exception of the following parameters: • red-flow-port • red-max-trans • red-sync-start-time • red-sync-comp-time • red-mgcp-port
Media Policy	
Media Profile	
MGCP	
Network Interface	
Net Management Control	
Network Parameters	The Network Parameters element is supported with the exception of the following parameters: • SCTP parameters
NTP Sync	
Packet Trace Config	
Q850 SIP Map	
Realm Config	

ACLI Configuration Elements	Parameter Details
Redundancy Config	The Redundancy Config element is supported with the exception of the following parameters: • state • port • cfg-port • cfg-max-trans • cfg-sync-start-time • cfg-sync-comp-time
Session Agent	
Session Group	
Session Router	
Session Constraints	
Session Translation	
SIP Config	The SIP Config element is supported with the exception of the following parameters: • red-sip-port • red-max-trans • red-sync-start-time • red-sync-stop-time
SIP Feature	
SIP Interface	collect>boot-state
SIP Manipulation	
SIP NAT	The SIP NAT element is supported with the exception of the following parameters: • ext-address
SIP Response Map	
SNMP	
Static Flow	
Steering Pool	
Surrogate Agent	
System	The System Config element is supported with the exception of the following parameters: • options
Test Pattern Rule	
Test Policy	
Test Translation	
TLS Global	

ACLI Configuration Elements	Parameter Details
TLS Profile	
Translation Rules	
Trap Receiver	

C

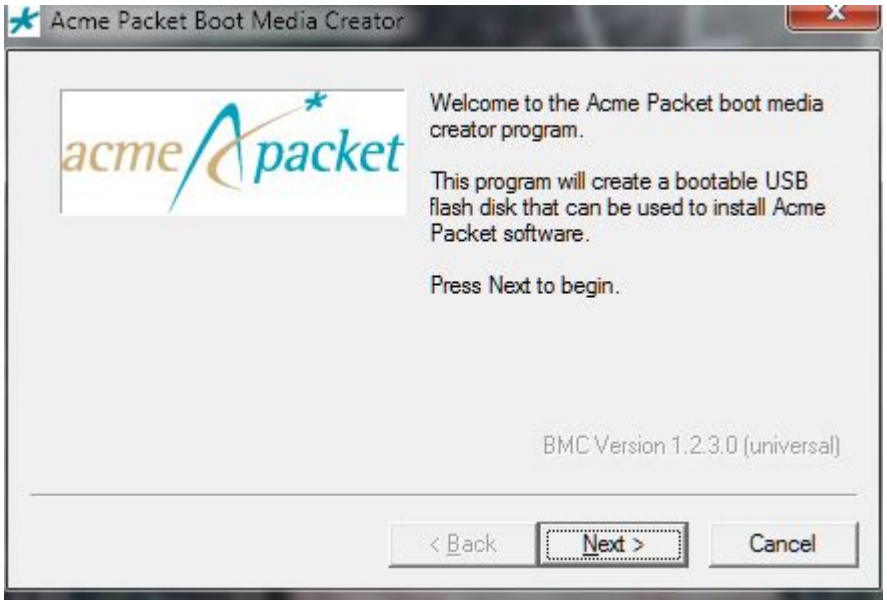
Boot Media Creator

Writing a Build Image

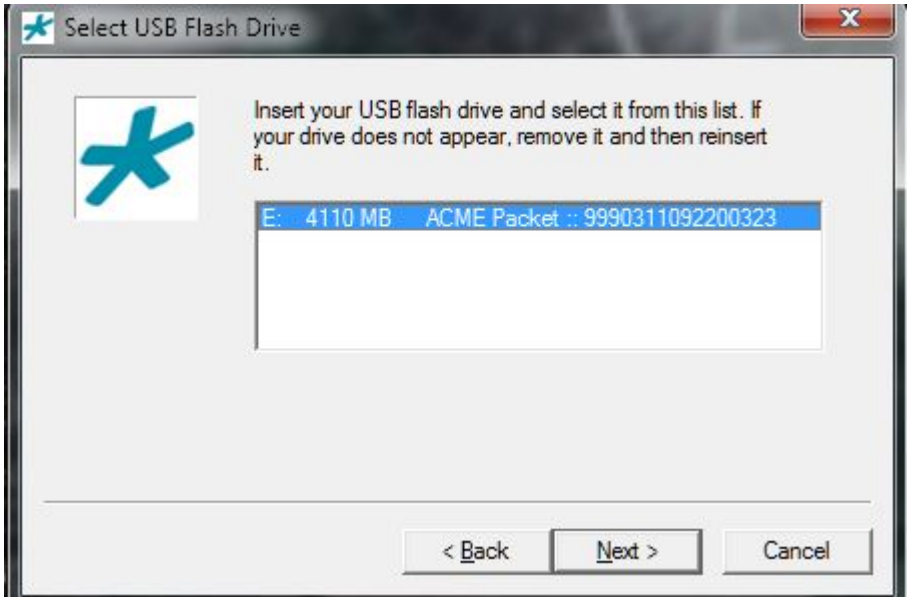
Channel partners will use the Boot Media Creator (BMC), a Windows executable file (nnSCz639-img-usb.exe for this release cycle) to write a build image to a USB stick. The USB stick will be subsequently be used to load the software to a server.

Use the following procedure to create a USB stick containing only a build image.

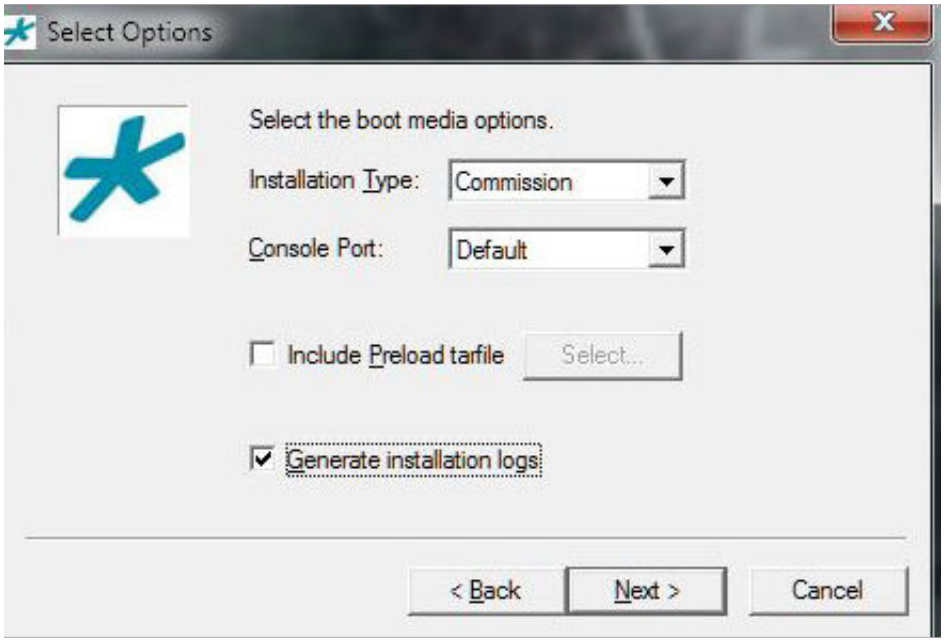
1. Open nnSCz639-img-usb.exe; click Next.



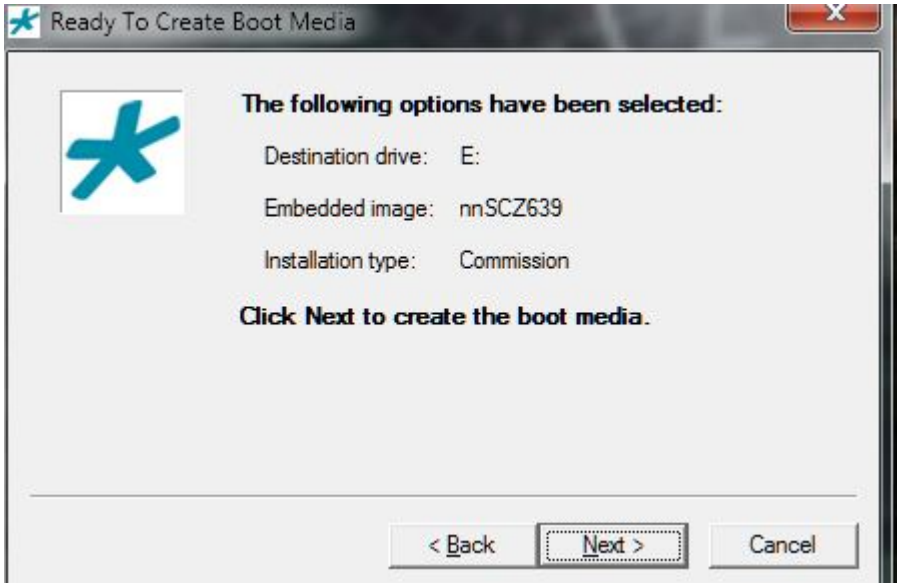
2. Insert the USB stick and/or select it from the displayed list; click Next.



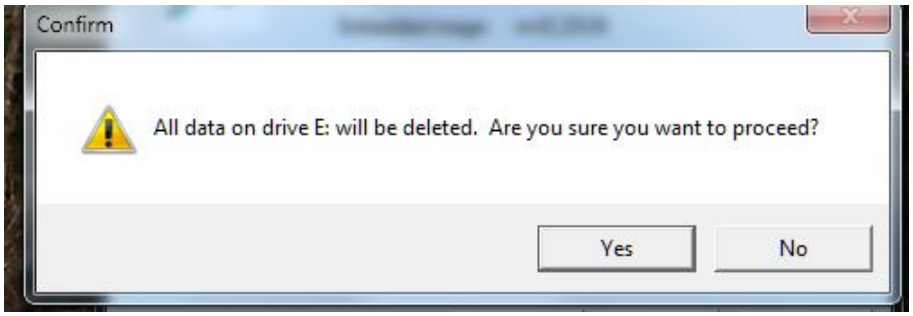
3. To write a build image to the USB stick, select Commission as the Installation Type, as shown below; click Next.



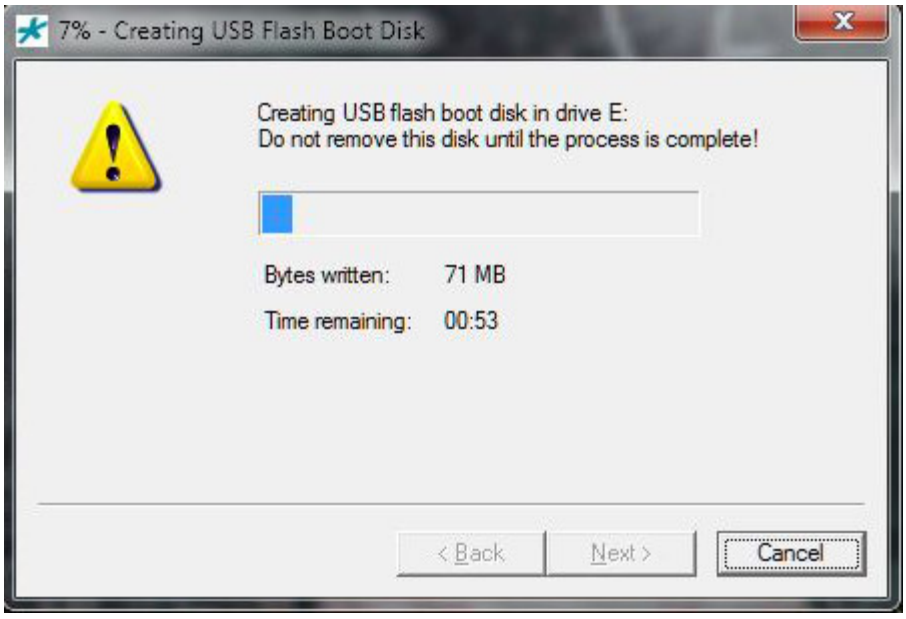
4. Confirm selected options; click Next.



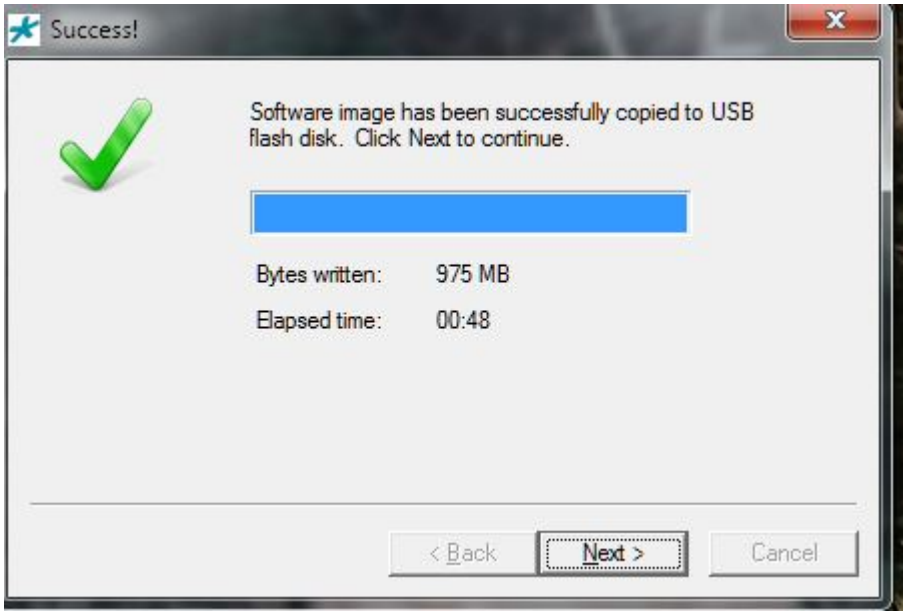
5. Heed the warning; click Yes.



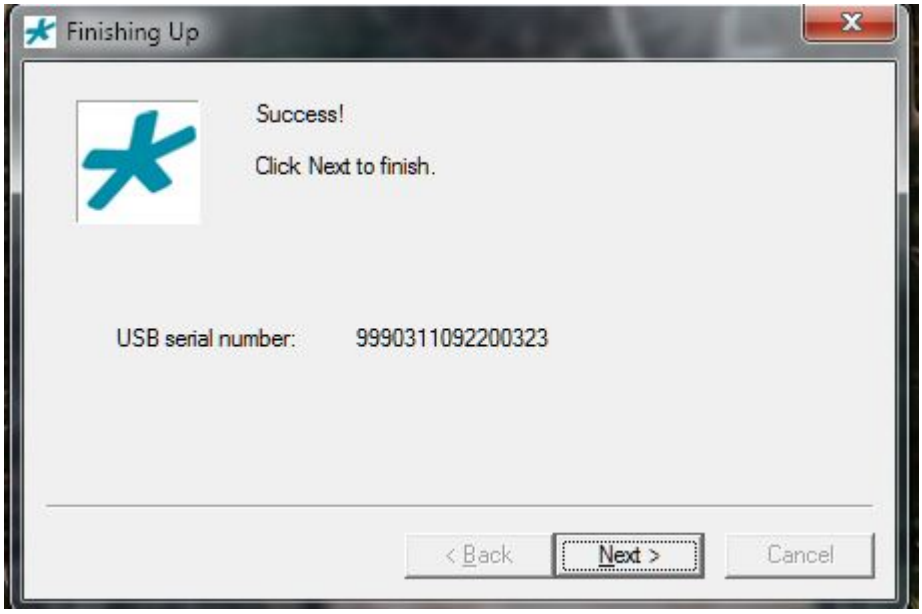
6. Wait while the BMC writes to the USB stick.



7. Click Next when the write operation completes.

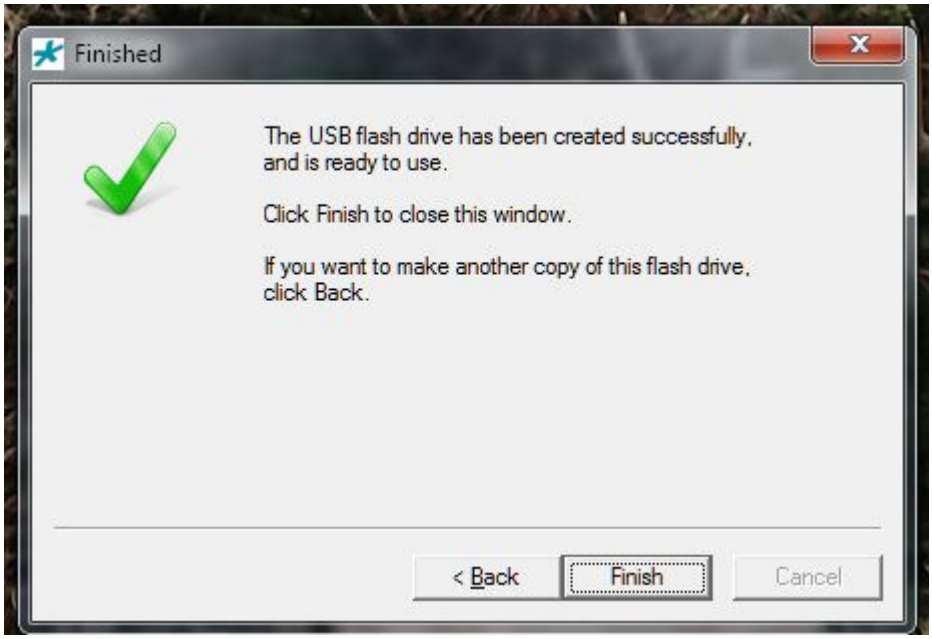


8. Click Next to finish this write operation.



Note: The displayed serial number, which identifies the USB stick, will be used by the end user to obtain product licenses.

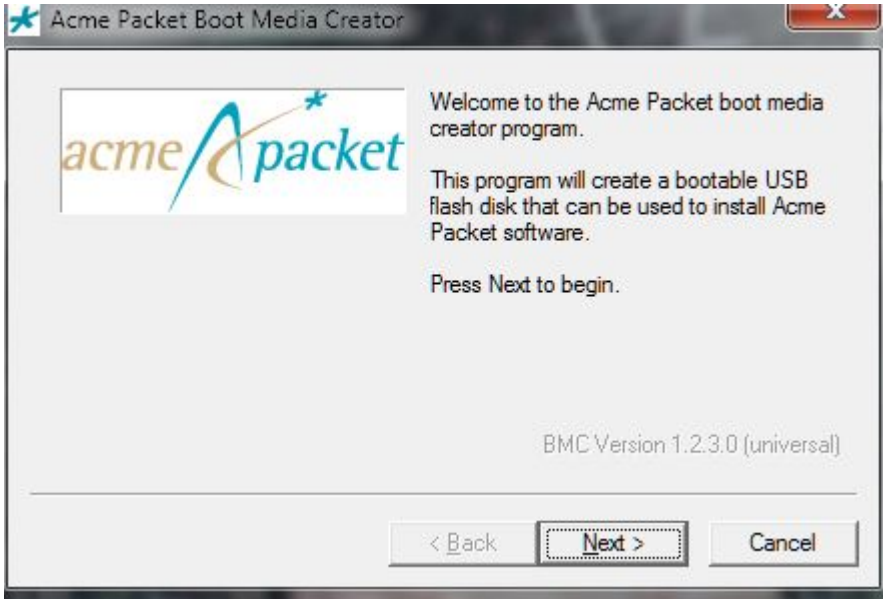
9. Click Back to make another copy, or Finish to exit the BMC.



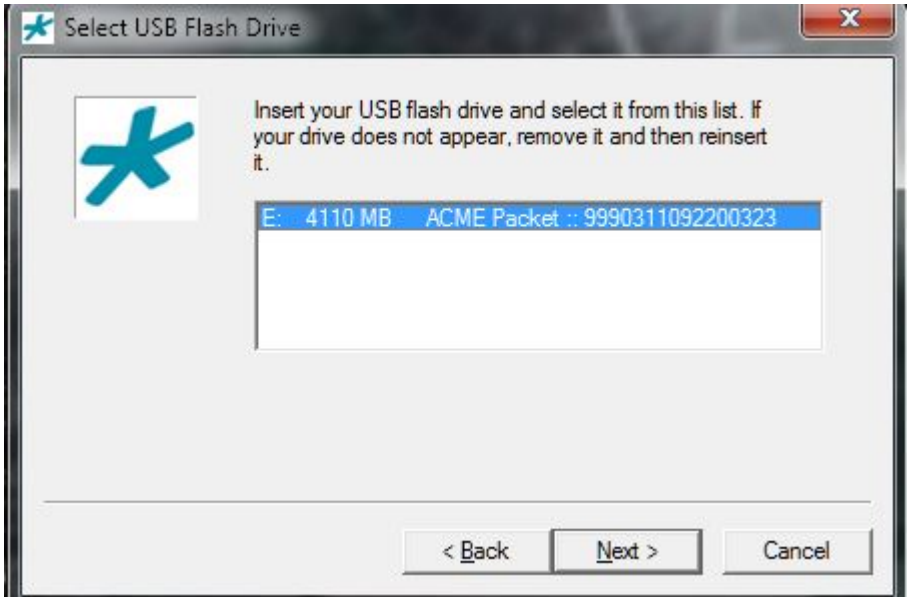
Writing a Build Image and .tar Archive

Use the following procedure to create a USB Stick containing both a build image and a pre-installed .tar archive.

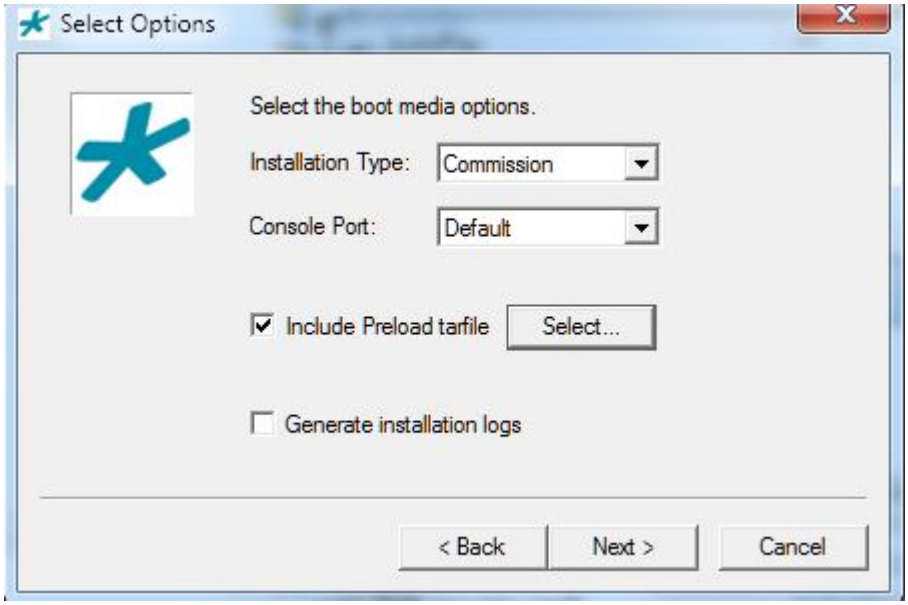
1. Open nnSCz639-img-usb.exe; click Next.



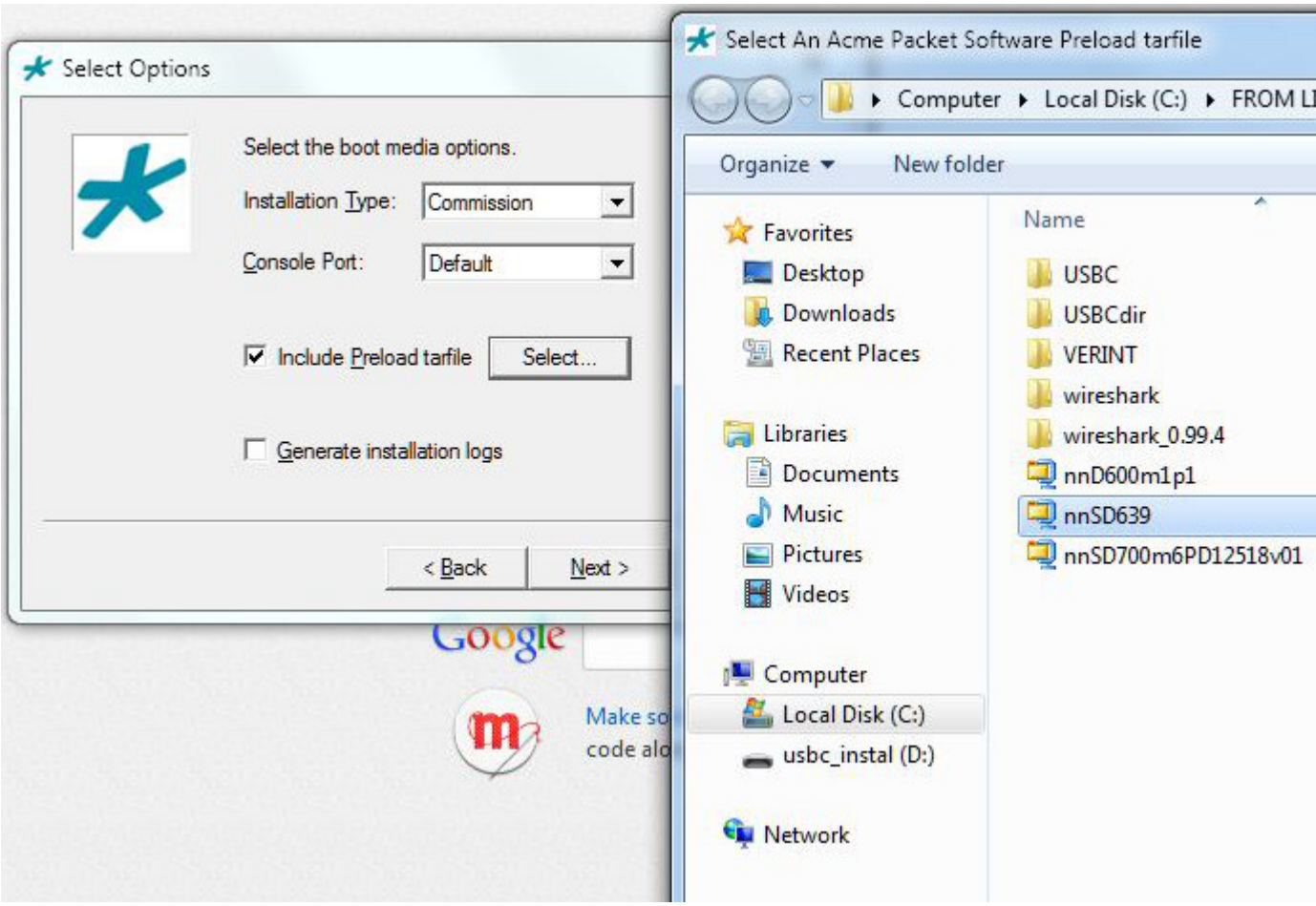
2. Insert the USB stick and/or select it from the displayed list; click Next.



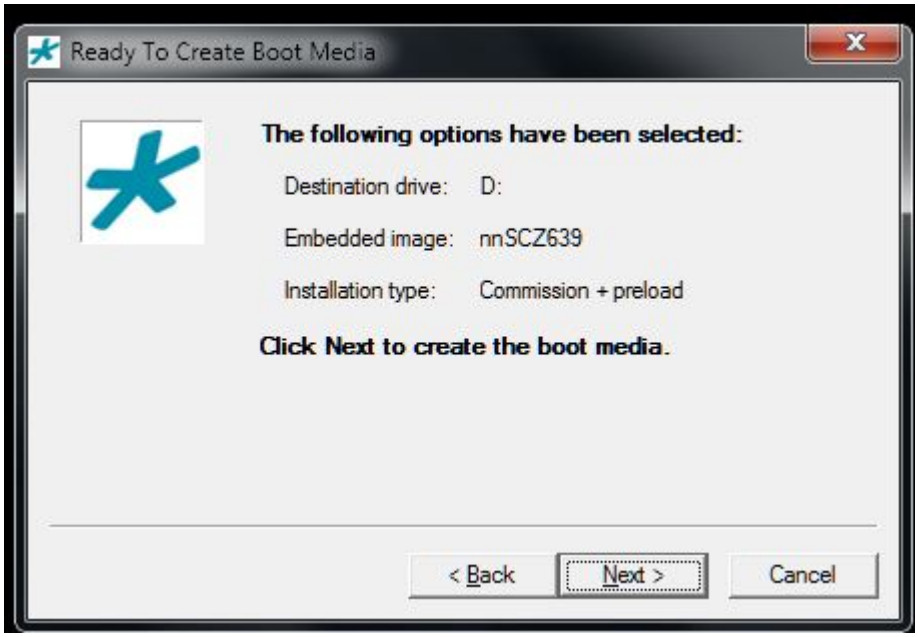
3. To write both a build image and a .tar archive to the USB stick, select Commission as the Installation Type, and click Include Preload tarfile as shown below.



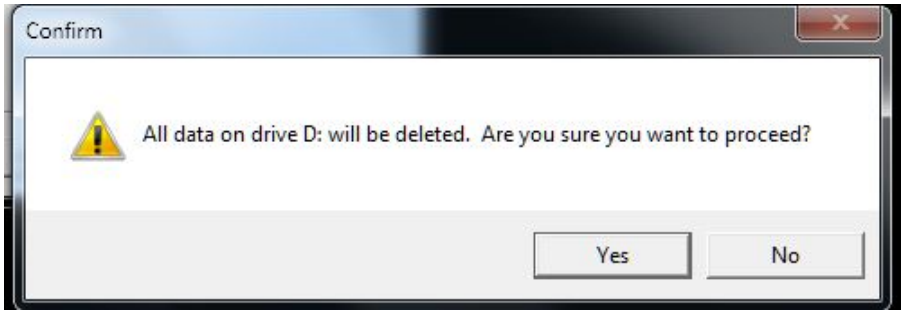
4. Use the Select button to navigate to the compressed archive to be written to the USB stick, select the archive, and click Next.



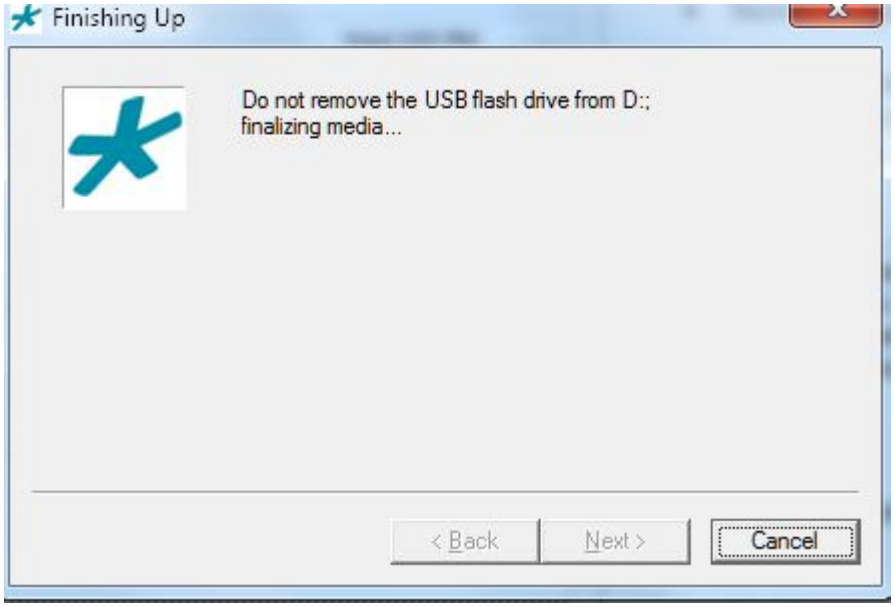
5. Confirm selected options; click Next.



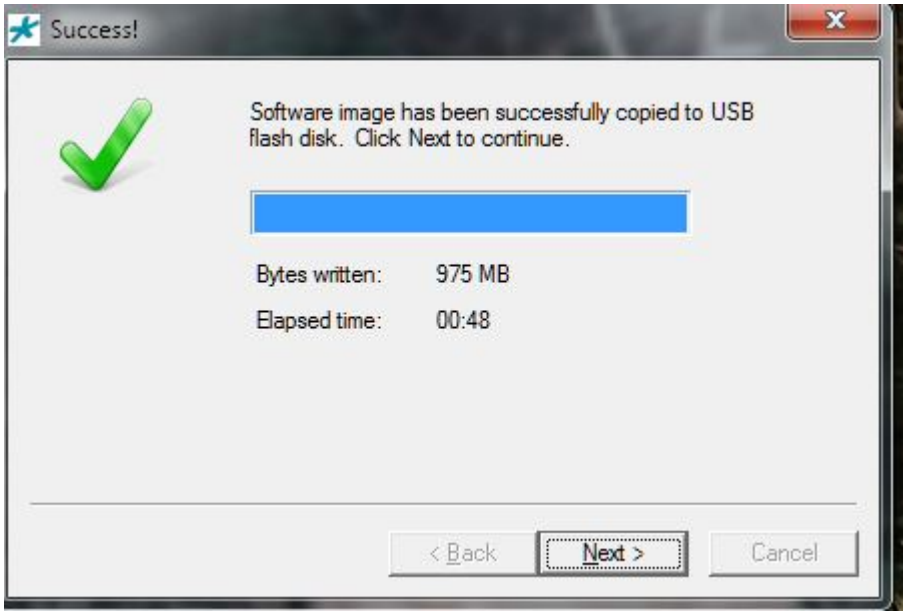
6. Heed the warning; click Yes.



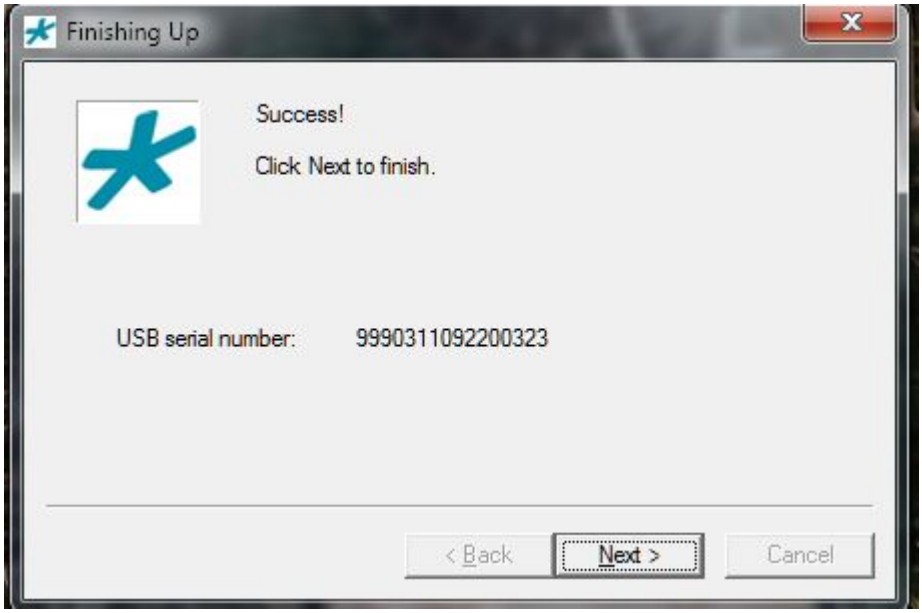
7. Wait while the BMC writes to the USB stick.



8. Click Next to complete this write operation.

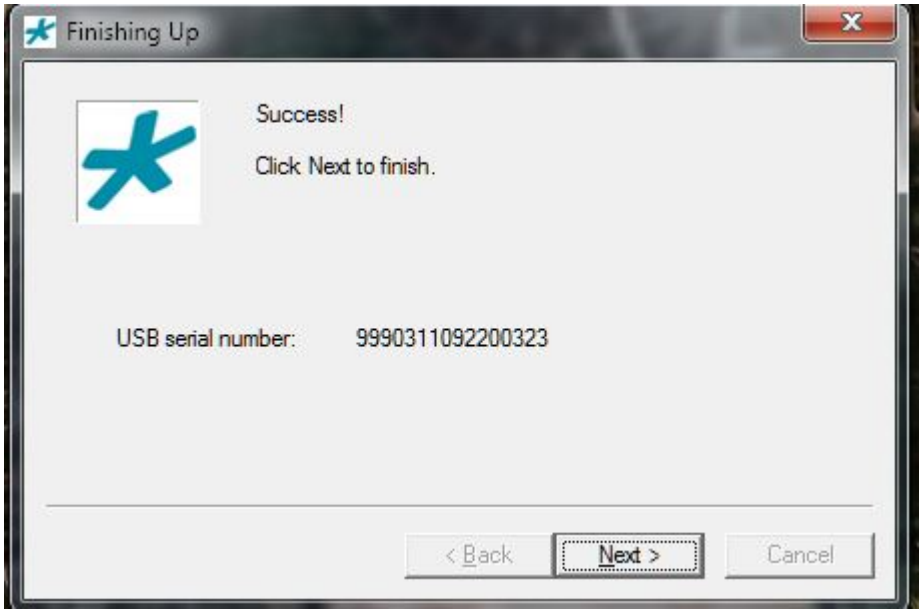


9. Click Next to finish this write operation.



Note: The displayed serial number, which identifies the USB stick, will be used by the end user to obtain product licenses.

10. Click Back to make another copy, or Finish to exit the BMC.



D

Configure the Web Server From the ACLI

You must configure and enable the Web server for Oracle Enterprise Session Border Controller (E-SBC) operations before you can use the Web GUI.

If you previously ran the Set Initial Configuration wizard from the Web GUI, confirm whether or not the Web GUI is already enabled.

The following procedure provides instructions to configure and enable the Web server through the ACLI.

Note: The Web GUI supports only IPv4.

To configure the Web server:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type system and press Enter to access the system-related objects.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type web-server-config and press Enter to access the event monitoring-related attributes.

```
ACMEPACKET(system)# web-server-config
ACMEPACKET(web-server-config)#
```

state—Enter whether or not to enable the Web GUI. Default is enabled. Valid values are:

- enabled
- disabled

```
ACMEPACKET(web-server-config)# state enabled
```

inactivity-timeout—Enter the amount of time, in minutes, that the Web GUI must have remained inactive before it ends Web session. For example, if the timeout value is set as 5, the Web session disconnects after 5 minutes of inactivity. Default is 5. Valid values are 0 to 20.

```
ACMEPACKET(web-server-config)# inactivity-timeout 5
```

Note: The following http-state, http-port, https-state, and https-port parameters may have already been set through the Web GUI installation wizard on the E-SBC. You can edit these parameters using the ACLI.

http-state—Enter whether or not to enable HTTP for accessing the Web server. Default is enabled. Valid values are:

- enabled

- disabled

```
ACMEPACKET(web-server-config)# http-state enabled
```

http-port—Enter the HTTP port to use to connect to the Web server. Default is 80. Valid values are 1 to 65535.

```
ACMEPACKET(web-server-config)# http-port 80
```

https-state—Enter whether or not to enable HTTPS (secure connection) for accessing the Web server. Default is disabled. Valid values are:

- enabled (default)
- disabled

```
ACMEPACKET(web-server-config)# https-state enabled
```

https-port—Enter the HTTPS port to use to connect to the Web server. Default is 443. Valid values are 1 to 65535.

```
ACMEPACKET(web-server-config)# https-port 443
```

tls-profile—Enter the Transport Layer Security (TLS) Protocol profile name to use with HTTPS. Default is blank. Valid values are alpha-numeric characters.

```
ACMEPACKET(web-server-config)# tls-profile tlsSM&T
```

Note: If you specify a tls-profile, and HTTP is enabled, the E-SBC checks against the TLS profile table for a match. If there is no match, the applicable errors display during the verification of the configuration. To create a TLS profile, see Chapter 22, Configuring a TLS Profile.

4. Enter exit to exit the Web server configuration.

```
ACMEPACKET(web-server-config)# exit
```

5. Enter exit to exit the system configuration.

```
ACMEPACKET(system)# exit
```

6. Enter exit to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

7. Enter save-config to save the configuration.

```
ACMEPACKET# save-config
```

8. Enter activate-config to activate as the current configuration.

```
ACMEPACKET# activate-config
```

E

Acronym List

General Use Acronyms

3GPP	3rd-Generation Partnership Project
A	
AAA	Authentication, Authorization, and Accounting
ACD	Automatic Call Distribution
ACL	Access Control List
ACLI	Acme Command Line Interface
ACP	Acme Control Protocol
ADMF	ADMinistration Function
AF	Access Function
AFID	Access Function Identifier
AIN	Advanced Intelligent Network
ALG	Application Layer Gateway
AM	Application Manager
ANI	Automatic Number Identification (ISDN)
ANSI	American National Standards Institute
AoR	Address of Record
AP	Application Protocol
API	Application Programming Interface
APN	Access Point Name
APPN	Advanced Peer-to-Peer Networking
ARP	Address Resolution Protocol
ARQ	Admission Request (H.323)

ASCII	American Standard Code for Information Interchange
ASIC	Application-Specific Integrated Circuit
ASN	Abstract Syntax Notation
ASN.1	Abstract Syntax Notation – 1
ASP	Application Service Provider, Active Server Pages, Adjunct Service Point
ASR	Access Service Request
ATCP	Async TCP
ATCP socket	Refers to a socket used for an async TCP connection.
ATCP stack	Refers to the separate TCP stack implemented in the ATCP task.
ATCP task	Refers to the task in the system in which the ATCP stack executes.
ATM	Asynchronous Transfer Mode
ATX	Advanced Technology Extended
B	
B2BGW	Back-to-Back Gateway
B2BUA	Back-to-Back User Agent
BBSRAM	Battery Backup Static Random Access Memory
BC	Bearer Capability
BCID	Billing Correlation Identifier
BER	Basic Encoding Rules
BGP	Border Gateway Protocol
BHCA	Busy Hour Call Attempts
BIOS	Basic Input/Output System
BIS	Bearer-Independent Setup
BITS	Building Integrated Timing Supply
B-ICI	Broadband Inter-carrier Interface (ATM)
B-ISUP	Broadband ISDN User Part
BNF	(augmented) Backus-Naur Form
BoD	Bandwidth on Demand
BoS	Bottom of Stack
bps	Bits per Second
BRAS	Broadband Remote Access Server
BRI	Basic Rate Interface (ISDN)
BSP	Board Support Package

BTU	British Thermal Units
C	
CA	Certificate Authority
CAC	Call Administration Control
CALEA	Communications Assistance to Law Enforcement Agencies
CAM	Content Addressable Memory
CARP	Cache Array Routing Protocol (to replace ICP)
CAS	Cordless Access Service
CBR	Constant Bit Rate (ATM)
CC	Country Code/ Call Content
CCC	Call Content Connection/ Handover Interface 3 (Call Content)
CCCid	Call Content Connection Identifier
CCM	Cisco Call Manager
CD	Call Data
CDC	Call Data Connection/ Handover Interface 2 (Call Data)
CDPN	Called Party Number
CDR	Call Detail Record
CE	Conformité Européenne (The CE marking is a European proof of conformity and is also described as "passport" that allows manufacturers and exporters to circulate products freely within the EU.)
CFM	Cubic Feet per Minute (fan speed)
CFU	Call Forwarding Unconditional
CGI	Common Gateway Interface
CIC	Carrier Indicator Code/Carrier Identification Code
CID	Caller Identification
CISSP	Certified Information Systems Security Professional
CLC	Close Logical Channel
CLCAck	Close Logical Channel Ack
CLEC	Competitive Local Exchange Carrier
CLI	Command-line Interface
Client-SI	Client-Server Information
CMS	Call Management Server
CNM	Customer Network Management
CO	Connection Oriented

CODEC	Coder/Decoder
CoS	Class of Service
CP	Communications Processor
CPL	Call Processing Language
CPLD	Complex Programmable Logic Device
CPM	Communications Processor Module
CPU	Central Processing Unit
CRA	Call Routing Apparatus
CRI	Call Related Information
CRLF	Carriage Return Line Feed
CS	Circuit Switch
CSA	Client Server Architecture
CSPDN	Circuit Switched Public Data Network
CSU	Channel Service Unit
CT	Cordless Telephone
CT-1	European analogue cordless telephone system
CT-2	Second-generation cordless telephone, Digital
CTI	Computer Telephony Integration
CUG	Closed User Group
D	
DA	Destination Address
DAM	Data Access Method; Data Asset Management
DDD	Direct Distance Dialing
DDF	Digital Distribution Frame
DECT	Digital European Cordless Telephone
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DIMM	Dual In-line Memory Module
DLCI	Data Link Connection Identifier
DLSR	Delay Since Last Send Report
DN	Directory Number
DNS	Domain Name Server/Service
DOM	Document Object Model

DoS	Denial of Service
DP	Destination Port
DPCM	Differential Pulse Code Modulation
DRAM	Dynamic Random Access Memory
DS	Differentiated Services
DSA	Digital Signature Algorithm
DSCP	DiffServ Codepoint
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSP	Digital Signal Processing
DSS	Digital Satellite System
DSU	Digital Service Unit
dTCP	Dynamic Transmission Control Protocol
DTD	Document Type Definition
dTLS	Dynamic Transport Layer Security
DTMF	Dial Tone Multi-Frequency
DWA	Device Watchdog Answer
DWR	Device Watchdog Request
E	
ED	Ending Delimiter
EEPROM	Electrically Erasable Programmable Read-Only Memory
EFTPOS	Electronics Funds Transfer Point of Sale
EGP	Exterior Gateway Protocol
EMB	Early Media Blocking
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EMS	Element Management System (Acme Packet)
ENUM	Refers to the use of an E.164 number, in reverse, with domain notation (i.e., dotted)
EPROM	Erasable Programmable Read-Only Memory
ER	Edge Router
ESD	Enterprise Session Director
ETSI	European Telecommunications Standards Institute
F	
FCC	Federal Communications Commission
FCP	Firewall Control Protocol

FEC	Forward Equivalence Class
FPGA	Field Programmable Gate Array
FQDN	Fully Qualified Domain Name
FS	Fast-start
FSA	Foreign SIP Agent (ACME-specific term?)
FTP	File Transfer Protocol
FTR	Flow Transform Record
G	
GA	Global Address
GB	Gigabyte
GBPS	Gigabits Per Second
GigE	Gigabit Ethernet
GK	Gatekeeper
GMII	Gig Media Independent Interface
GNU	GNUs not UNIX
GOSIP	Government Open Systems Interconnection Profile
GRUU	Globally Routable User Agent URI
GPS	Global Policy Server/Global Positioning System
GSA	Global System Administrator
GSM	Global Systems for Mobile Communications
GSTN	Global Switched Telephone Network
GW	Gateway
H	
HA	High Availability (Acme Packet redundancy solution)
HNT	Hosted NAT Traversal (Acme Packet)
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
I	
IAD	Integrated Access Device
IANA	Internet Assigned Numbers Authority
ICE	In Circuit Emulator
ICMP	Internet Control Message Protocol
ICP	Internet Cache Protocol
I-CSCF	Interrogating Call Session Control Function
IDS	Intrusion Detection System

IE	Information Element
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IIS	Internet Information Server
IKE	Internet Key Exchange
ILEC	Independent Local Exchange carrier (USA)
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IN	Intelligent Network
I/O	Input/Output
IOS	Internetworking Operating System
IP	Internet Protocol (IPv4, IPv6)
IPC	Inter-process Communication
IPDR	Internet Protocol Data Record
IPSec	Internet Protocol Security
IPtel	Internet Protocol Telephony
IPv	Internet Protocol version
IS	Intercept Server
ISDN	Integrated Services Digital Network
ISO	International Organization of Standardization
ISP	Internet Service Provider
ITAD	Internet Telephony Administrative Domain
ITSP	Internet Telephony Service Provider
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
IVR	Interactive Voice Response
IWF	Interworking Function (referring to the Net-Net ESD's SIP-H.323 interworking)
IXC	Interexchange Carrier
J	
JTAG	Joint Test Action Group
JTAPI	Java telephony application programming interface
K	

Kb	Kilobits
KB	Kilobytes
Kbps	Kilobits per second
KEA	Key Exchange Algorithm
KTS	Key Telephone System
L	
LA	Local Address
LAES	Lawfully Authorized Electronic Surveillance
LAN	Local Area Network
LATA	Local Access and Transport Area
LCD	Liquid Crystal Display
LDAP	Lightweight Direct(ory) Access Protocol
LEA	Law Enforcement Agency
LEAF	Law Enforcement Administrative Function
LEC	Local Exchange Carrier
LED	Light Emitting Diode
LEM	Local Element Manager (Acme Packet)
LEN	Local Exchange Node
LNP	Local Number Portability
LRT	Local Routing Table
LOS	Loss of Signal
LS	Location Server
LSB	Least Significant Bit
LSR	Label-switching router
M	
MAC	Media Access Control/ Message Authentication Code
MAN	Metropolitan Area Networks
Mb	Megabits
MB	Megabytes
MBCD	Middlebox Control Daemon (Acme Packet)
Mbone	Multicast Backbone
MC	Monitoring Center
MCU	Multi-party Conference Unit
MD5	Message Digest 5 (hash function)
MF	Media Firewall

MG	Media Gateway
MHz	Megahertz
MIB	Management Information Base
MIB	—Management Information Base II
MIBOC	Middlebox Control Protocol
MIDCOM	Middle Box Communications
MIKEY	Multimedia Internet Keying
MIME	Multipurpose Internet Mail Extension
MOC	Mandatory, Optional, Conditional
MoIP	Messaging over Internet Protocol
MP	Main Processor
µP	Microprocessor (subsystem)
MPLS	Multi-protocol Label Switching
MR	Media Router
MRCP	Media Router Control Protocol
MSB	Most Significant Bit
MSD	Master-Slave Determination
MTA	Message Transfer Agent / Multimedia Terminal Adapter
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
MTU	Maximum Transmission Unit
MX	Mail Exchange
N	
N-ACD	Network Automotive Call Distribution
NANP	North American Numbering Plan
NAPT	Network Address Port Translation
NAPTR	Naming Authority Pointer
NAS	Network Access Security
NAT	Network Address Translation
Nco	Network Code of Practice
NCP	Network Control Point
NEBS	Network Equipment - Building Systems/Standards
NE	Network Element
NIC	Network Interface Card
NMS	Network Management Station

NP	Network Processor
NPU	Network Processing Unit
NSRG	Network Signaling Record Generator
NTE	Networking Terminating Equipment
NTP	Network Time Protocol
NTU	Networking Terminating Unit
NVRAM	Non-volatile Random Access Memory
O	
OAM	Operation, Administration, and Maintenance
OC	Optical Carrier
OC	n—Optical Carrier transport
OCSP	Online Certificate Status Protocol
OCx	Optical Carrier level
OEI	Optical Electrical Interface
OEM	Original Equipment Manufacturer
OID	Object Identifier
OLC	Open Logical Channel
OLCAck	Open Logical Channel Ack
ONP	Open Network Provision
OS	Operating System
OSP	Open Settlement Protocol
OSPF	Open Shortest Path First
OSS	Operations Support Systems
P	
PABX	Private Automatic Branch Exchange
PAC	Performance, Availability, Capacity (Acme Packet)
PACS	Personal Access Communications Systems
PAT	Port Address Translation
PBX	Private Branch Exchange
P-CSCF	Proxy-Call Session Control Function
PCB	Printed Circuit Board
PCDATA	Parseable Data Characters
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PCN	Personal Communications Network

PCS	Personal Communications Services
PD	Packet Data
PDCS	Packet Cable Distributed Call Signaling
PDH	Plesiochronous Digital Hierarchy
PDN	Public Data Network / Packet Data Network
PDP	Policy Decision Point
PDU	Protocol Data Unit (or Packet Data Unit)
PEM	Privacy Enhanced Mail
PEP	Policy Enforcement Point/Protocol Extensions Protocol
Perl	Practical Extraction Report Language
PHY	Physical Layer Device
PIB	Policy Information Base
PING	Packet Internet Groper
PINT	PSTN and IP Internetworking
PKCS-7	RFC 2315, Cryptographic Message Syntax, Version 1.5
PKCS-10	RFC 2314, Certificate Request Syntax, Version 1.5
PKI	Public Key Infrastructure
PMC	PCI Mezzanine Card
PNNI	Private Network Node Interface (ATM)
PNO	Public Network Operator
POP	Point of Presence
POS	Packet Over SONET
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PROM	Programmable Read-Only Memory
PS	Policy Server
PSAP	Public Safety Answering Point
PSTN	Public Switched Telephone Network (Telecom Network)
PTE	Packet Transform Engine
PTO	Public Telecommunications Operator
PTT	Post, Telephone, and Telegraph
PWB	Printed Wiring Board
Q	
QoS	Quality of Service
QSIG	Unified International Digital Corporate Network Signaling Standard

R	
RACF	Resource and Admission Control Function
RADIUS	Remote Authentication Dial-in User Service
RAM	Random Access Memory
RARP	Reverse Address Resolution Protocol
RAS	Remote Access Service; Registration Admission and Status (H.323)
RC	Registration Cache
RC2 and RC4	Rivest encryption ciphers developed for RSA Data Security
RED	Random Early Discard
REN	Ringer Equivalent Number
RFC	Request for Comments
RIP	Routing Information Protocol
RISC	Reduced Instruction Set Chip
RMON	Remote (Network) Monitoring
ROM	Read-Only Memory
RPC	Remote Procedure Call
RR	Received Report
RS-232	Recommended Standard 232 (computer serial interface, IEEE)
RSA	Rivest, Shamir, & Adleman (public key encryption technology)
RSIP	ReStart In Progress
RSVP	Resource Reservation Protocol
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
RTP/AVP	Real-time Transport Protocol/Audio-Video Protocol
RTSP	Real-time Streaming Protocol
RTT	Round Trip Time
S	
SA	Source Address / Session Agent / Security Association
SAG	Session Agent Group
SBC	Session Border Controller
SCE	Service Control Environment
SCP	Service Control Point
S-CSCF	Serving Call Session Control Function

SCTP	Streaming Control Transmission Protocol
SDES	Source Description RTCP (Real-Time Control Protocol) Packet
SDH	Synchronous Digital Hierarchy
SDP	Session Description Protocol
SDRAM	Synchronous Dynamic Random Access Memory
SERDES	Serial De-serializer
SFE	Security Front End
SHA-1	Secure Hash Algorithm, a hash function used by the U.S. Government
SIG	Special Interest Group
SIM	Subscriber Identity Module
SIMM	Single In-line Memory Module
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SME	Small to Medium Enterprise(s)
SMIL	Synchronized Multimedia Integration Language
SMP	Simple Management Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOCKS	SOCKetS server
SONET	Synchronous Optical Network
SP	Source Port / Service Provider
SR	(Net-Net) Session Router
SRAM	Static Random Access Memory
SRS	Session Routing System
SRTP	Secure Real-Time Transport Protocol
SRV	Resource record for servers (DNS)
SS	Slow-start
SS7	Signaling System 7
SSH	Secured Shell or Secure Socket Shell
SSL	Secure Socket Layer
SSP	Service Switching Point
sTCP	Static Transmission Control Protocol
STL	Standard Template Library
sTLS	Static Transport Layer Security

STP	Signal Transfer Point; Service Transfer Point
SVC	Signaling Virtual Channel (ATM) / Switched Virtual Circuit (Packet Switching)
T	
TA	Terminal Adapter (ISDN)
TAC	Terminal Access Control
TACACS+	Terminal Access Controller Access Control System
TAPI	Telephony Application Program Interface
TAXI	Transparent Asynchronous Transmitter/Receiver Interface
TCB	Task Control Bar/Task Control Block
TCI	Tag Control Identifier
TCP	Transmission Control Protocol
TCP	IP—Transmission Control Protocol /Internet Protocol
TCS	Terminal Capability Set
TDM	Time Division Multiplexing
TEN	Transit Exchange Node
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security (same as SSL)
TLV	Tag Length Value
TM	Traffic Manager
TMN	Telecommunications Management Network
ToS	Type of Service
TRIB	Telephony Routing Information Base
TRIP	Telephony Routing over IP
TS	Time Slot
TSAP	Transport Service Access Point
TSAPI	Telephony Server API
TTL	Time to Live
TTR	Time to Resume
U	
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UE	User Equipment

UL	Underwriters Laboratories
UMTS	Universal Mobile Telecommunications Systems
UNI	User-to-Network Interface
UPS	Uninterruptible Power Supply
UPT	Universal Portable Telephone
URI	User Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTP	Unshielded Twisted Pair
V	
VAC	Volts Alternating Current
VANS	Value Added Network Services
VAR	Value Added Reseller
VarBind	Variable Binding
VBR	Variable Bit Rate
VC	Virtual Channel (ATM)/Virtual Container (SDH)
VCC	Virtual Channel Connection (ATM)
VCI	Virtual Channel Identifier
VDC	Volts Direct Current
VFD	Vacuum Florescent Display
VLAN	Virtual Local Area Network
VLL	Virtual Leased Lines
VoIP	Voice Over Internet Protocol
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VSA	Vendor-specific Attribute (RADIUS extension)
VTOA	Voice and Telephony over ATM
W	
WAN	Wide Area Network
WLL	Wavelength Division Multiplex
X	
XE	Translation Engine
XML	Extensible Markup Language

XSM	External Search Machine
Y (None to list.)	
Z (None to list.)	

F

Advanced Logging

Advanced Logging allows targeted logging by over riding log levels, so that only a specific SIP request and its related messages are logged. The system matches criteria that you configure to determine which requests to log. The system also logs all messages related to the request, such as any responses, in-dialog messages, media, timers, and so on. Advanced Logging supports multiple matching criteria for incoming requests and rate limiting. Advanced log files are smaller than debug files because the system logs only the specified number of matches in the specified period of time. Since the files are smaller, Advanced Logging uses fewer system resources than debug logging. To make searching easier, the system labels each log.

You can deploy advanced logging by one or both of the following methods.

- Configure mode. Define sip-advanced-logging under session-router. This method reconfigures the system and the configuration persists after a system reboot.
- Command line. From the Advanced Logging SPL plug-in that is included in the software, you can enable, start, and stop advanced logging without changing the system configuration. When configured from the command line, advanced logging does not persist after a system reboot.

Note: Configure mode and Command Line are separate deployment methods that do not depend on each other or affect each other.

The system provides the following options for configuring the scope of advanced logging.

- Request-only. Logs only the matched message.
- Transaction. Logs only the request and the response.
- Session. Logs the matched message and anything else related to the session.
- Session and Media. Logs the matched message, anything related to the session, and media.

The system provides the following options for configuring the advanced logging criteria.

- Received Session-Agent. By IP address or hostname
- Request Type. Such as INVITE vs. SUBSCRIBE
- Received Realm Name.
- Request URI. User and host. Limited to 2 condition entries, when using both types.
- To header. User and host. Limited to 2 condition entries, when using both types.
- From header. User and host. Limited to 2 condition entries, when using both types.
- Call-id. Matches the Call-id header.
- Rate Limiting. By specified number of matched requests over a specified period of time.
- Scope of Logging. Options include Request Only, Transaction, All Relating to Session, All Relating to Session and Media.

Enable Advanced Logging - Command Line

You can enable advanced logging and set the log matching criteria from the command line by way of the AdvancedLogging.lua SPL-plugin. When adding log matching criteria, note that within in each set of criteria:

- an AND relationship means that all conditions must match before the system generates the log.
- an OR relationship means that only one set of conditions must match before the system generates the log.

Note: The system does not require you to **save** and **activate** after performing this procedure.

Procedure

1. Use the spl start sip advanced-logging command to enable advanced logging.
2. Use the following commands, as needed, to configure advanced logging.

Command	Description
spl set sip advanced-logging add-criteria	Adds another set of matching criteria.
spl set sip advanced-logging log-label <label string>	Any logs of requests that are matched will have the specified <label string> appended before each log message for easier searching.
spl set sip advanced-logging rate-count <match count>	Sets the rate-limiting to log only <match count> number of matching requests per time window.
spl set sip advanced-logging rate-time <time window>	Sets the rate-limiting time window in seconds.
spl set sip advanced-logging in-agent <session-agent>	Adds to the current set of matching criteria that the request must come from the specified incoming session-agent hostname.
spl set sip advanced-logging in-realm <realm-id>	Adds to the current set of matching criteria that the request must come from the specified incoming realm identifier.
spl set sip advanced-logging request-type <method name>	Adds to the current set of matching criteria that the request must be of the specified request method type, for example, INVITE and REGISTER.
spl set sip advanced-logging from-uri-host <FROM URI host portion>	Adds to the current set of matching criteria that the request FROM headerURI host portion must match the specified value exactly.
spl set sip advanced-logging from-uri-user <FROM URI username portion>	Adds to the current set of matching criteria that the request FROM headerURI username portion must match the string and the specified value exactly.
spl set sip advanced-logging to-uri-host <TO URI host portion>	Adds to the current set of matching criteria that the request TO headerURI host portion must match the string and the specified value exactly.
spl set sip advanced-logging to-uri-user <TO URI username portion>	Adds to the current set of matching criteria that the request TO headerURI username portion must match the string and the specified value exactly.
spl set sip advanced-logging request-uri-host <RURI host portion>	Adds to the current set of matching criteria that the request RURI headerURI host portion must match the string and the specified value exactly.
spl set sip advanced-logging request-uri-user <RURI username portion>	Adds to the current set of matching criteria that the request RURI headerURI username portion must match the string and the specified value exactly.

Command	Description
spl set sip advanced-logging header <header-type> <header-value>	Adds to the current set of matching criteria that the request must have a header of type <header-type> with a value of <header-value> with exact string matches.

Enable Advanced Logging - Configure Mode

From Configure mode, define sip-advanced-logging and advanced-log-condition. The criteria that you configure remaps the message logging and modifies the system configuration. You must save and activate these changes to the configuration.

When configuring multiple sip-advanced-logging configurations, note the following.

- The system evaluates each configuration individually in an OR relationship.
- The system evaluates all conditions and they must all match in an AND relationship.

1. From Configure Mode, go to session-router > sip-advanced-logging and configure the following.
   - Name. Name to display on the log message for this set of criteria.
   - Level. Type one: zero, none, emergency, critical, major, minor, warning, notice, info, trace, debug, or detail.
   - Scope. Type one: request-only, transaction, session, or session-and-media.
   - Matches-per-window. Type a number between 1 and 999999999.
   - Window-size. Type a number between 1 and 999999999.
   - Conditions. Type AdvancedLogCondition.
2. From Configure Mode, go to session-router > sip-advanced-logging > advanced-log-condition and configure the following.
   - Match-type. Type one or more with either and or or between items : request-type, recv-agent, recv-realm, request-uri-user, request-uri-host, to-header-user, to-header-host, from-header-user, from-header-host, or call-id.
   - Match-value. Type the string that you want to match the incoming message. For example, to match "To-header-user" to the value 1234@<companyname>.com, type 1234.
3. Exit, save, and activate.

Disable Advanced Logging - Command Line

Confirm that advanced logging is enabled by way of the AdvancedLogging.lua SPL-plugin.

You can disable advanced logging by way of the command line without affecting any sip-advanced-logging that is enabled by way of the Configure Mode.

From the AdvancedLogging.lua SPL-plugin, run the spl stop advanced-logging command.

Disable Advanced Logging - Configure Mode

Confirm that Advanced Logging is enabled in Configure Mode.

To disable Advanced Logging in Configure Mode, clear all of the settings for sip-advanced-logging and advanced-log-condition. Disabling Advanced Logging in Configure Mode does not affect Advanced Logging that is enabled from the command line.

1. From Configure Mode, go to session-router > sip-advanced-logging and clear the following information.
   - Name
   - Level

- Scope
- Matches-per-window
- Window-size
- Conditions

2. From Configure Mode, go to session-router > sip-advanced-logging > advanced-log-condition and clear the following information.

   - Match-type
   - Match-value

3. Exit, save, and activate.

Clear Advanced Logging Criteria - Command Line

Use this procedure to clear the advanced logging conditions that were configured from the command line. This procedure clears all conditions and returns the system to the default state, in which all requests are matched.

From the AdvancedLogging.lua SPL-plugin, run the spl set sip advanced-logging clear-matching command.

View Advanced Logging Status - Command Line

View the status of advanced logging to see its state, configuration criteria, and count data.

Procedure

From the AdvancedLogging.lua SPL-plugin, run the spl show sip advanced-logging command.
The system displays the following information.

- State
- Log Label
- Rate Limit
- Matching Criteria
- Match Count
- Logged Count

G

TACACS+ AAA

TACACS+ (Terminal Access Controller Access Control System Plus) is a protocol originally developed by Cisco Systems, and made available to the user community by a draft RFC, *TACACS+ Protocol, Version 1.78* (draft-grant-tacacs-02.txt). TACACS+ provides AAA (Authentication, Authorization, and Accounting) services over a secure TCP connection using Port 49.

TACACS+ Introduction

Like DIAMETER and RADIUS, TACACS+ uses a client/server model in which a Network Access Server (NAS) acts in the client role and a TACACS+ equipped device (a daemon in TACACS+ nomenclature) assumes the server role. For purposes of the current implementation, the Oracle Oracle Enterprise Session Border Controller functions as the TACACS+ client. Unlike RADIUS, which combines authentication and authorization, TACACS+ provides three distinct applications to provide finer grade access control.

Authentication is the process that confirms a user's purported identity. Authentication is most often based on a simple username/password association, but other, and more secure methods, are becoming more common. The following authentication methods are support by the current implementation: simple password, PAP (Protocol Authentication Protocol), and CHAP (Challenge Handshake Authentication Protocol).

Authorization is the process that confirms user privileges. TACACS+ can provide extremely precise control over access to system resources. In the current implementation, TACACS+ controls access to system administrative functions.

TACACS+ provides secure communication between the client and daemon by encrypting all packets. Encryption is based on a shared-secret, a string value known only to the client and daemon. Packets are encrypted in their entirety, save for a common TACACS+ header.

The cleartext header contains, among other fields, a version number, a sequence number. and a session ID. Using a methodology described in Section 5 of the TACACS+ draft RFC, the sender encrypts outbound cleartext messages by repetitively running the MD5 hash algorithm over the concatenation of the session ID, shared-secret, version number, and sequence number values, eventually deriving a virtual one-time-pad of the same length as the message body. The sender encrypts the cleartext message with an XOR (Exclusive OR) operation, using the cleartext message and virtual one-time-pad as inputs.

The message recipient, who possesses the shared-secret, can readily obtain the version number, sequence number, session ID, and message length from the cleartext header. Consequently, the recipient employs the same methodology to derive a virtual one-time-pad identical to that derived by the sender. The recipient decrypts the encrypted message with an XOR operation, using the encrypted message and virtual one-time-pad as inputs.

Details on TACACS+ are available in the ACLI Configuration Guide.

The TACACS+ implementation is based upon the following internet draft.

draft-grant-tacacs-02.txt, *The TACACS+ Protocol Version 1.78*

Other relevant documents include

RFC 1321, *The MD-5 Message Digest Algorithm*

RFC 1334, *PPP Authentication Protocols* .

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

TACACS+ Authentication

The Oracle Enterprise Session Border Controller uses TACACS+ authentication services solely for the authentication of user accounts. Administrative users must be authenticated locally by the Oracle Enterprise Session Border Controller.

The current TACACS+ implementation supports three types of user authentication: simple password (referred to as ascii by TACACS+), PAP, and CHAP.

ascii Login

ascii login is analogous to logging into a standard PC. The initiating peer is prompted for a username, and, after responding, is then prompted for a password.

PAP Login

PAP is defined in RFC 1334, *PPP Authentication Protocols*. This protocol offers minimal security in that passwords are transmitted as unprotected cleartext. PAP login differs from ascii login in that the username and password are transmitted to the authenticating peer in a single authentication packet, as opposed to the two-step prompting process used in ascii login.

CHAP Login

CHAP is defined in RFC 1994, *PPP Challenge Handshake Authentication Protocol.* CHAP is a more secure than PAP in that it is based on a shared-secret (known only to the communicating peers), and therefore avoids the transmission of cleartext authentication credentials. CHAP operations can be summarized as follows.

After a login attempt, the initiator is tested by the authenticator who responds with a packet containing a challenge value — an octet stream with a recommended length of 16 octets or more. Receiving the challenge, the initiator concatenates an 8-bit identifier (carried within the challenge packet header), the shared-secret, and the challenge value, and uses the shared-secret to compute an MD-5 hash over the concatenated string. The initiator returns the hash value to the authenticator, who performs the same hash calculation, and compares results. If the hash values match, authentication succeeds; if hash values differ, authentication fails.

Authentication Message Exchange

All TACACS+ authentication packets consist of a common header and a message body. Authentication packets are of three types: START, CONTINUE, and REPLY.

START and CONTINUE packets are always sent by the Oracle Enterprise Session Border Controller, the TACACS+ client. START packets initiate an authentication session, while CONTINUE packets provide authentication data requested by the TACACS+ daemon. In response to every client-originated START or CONTINUE, the daemon must respond with a REPLY packet. The REPLY packet contains either a decision (pass or fail), which terminates the authentication session, or a request for additional information needed by the authenticator.

TACACS+ Header

The TACACS+ header format is as follows.

```
+----+----+--------+--------+--------+
|maj |min | type   | seq_no | flags  |
|ver |ver |        |        |        |
+----+----+--------+--------+--------+
| session_id                          |
+-------------------------------------+
| length                              |
+-------------------------------------+
```

maj ver

This 4-bit field identifies the TACACS+ major protocol version, and must contain a value of 0xC .

min ver

This 4-bit field identifies the TACACS+ minor protocol version, and must contain either a value of 0x0 (identifying TACACS+ minor version 0) or a value of 0x1 . (identifying TACACS+ minor version 1). Minor versions 0 and 1 differ only in the processing of PAP and CHAP logins.

type

This 8-bit field identifies the TACACS+ AAA service as follows:

0x1 — TACACS+ Authentication

0x2 — TACACS+ Authorization

0x3 — TACACS+ Accounting

sequence-no

This 8-bit field contains the packet sequence for the current session.

The first packet of a TACACS+ session must contain the value 1; each following packet increments the sequence count by 1. As TACACS+ sessions are always initiated by the client, all client-originated packets carry an odd sequence number, and all daemon-originated packets carry an even sequence number. TACACS+ protocol strictures do not allow the sequence_no field to wrap. If the sequence count reaches 255, the session must be stopped and restarted with a new sequence number of 1.

flags

This 8-bit field contains flags as described in Section 3 of the draft RFC; flags are not under user control.

session_id

This 32-bit field contains a random number that identifies the current TACACS+ session — it is used by clients and daemons to correlate TACACS+ requests and responses.

length

This 32-bit field contains the total length of the TACACS+ message, excluding the 12-octet header — in other words, the length of the message body.

Authentication START Packet

The Oracle Enterprise Session Border Controller, acting as a TACACS+ client, sends an authentication START packet to the TACACS+ daemon to initiate an authentication session. The daemon must respond with a REPLY packet.

The authentication START packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|--------+--------+--------+--------+
```

```
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
+--------+--------+--------+--------+
|               user ...             |
+------------------------------------+
|               port ...             |
+------------------------------------+
|             rem-addr ...           |
+------------------------------------+
|               data ...             |
+------------------------------------+
```

action

This 8-bit field contains an enumerated value that identifies the requested authentication action. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level requested by an authenticating user. For the current TACACS+ authentication implementation, this field always contains a value of 0x01 , indicating the user level.

authen-type

This 8-bit field contains an enumerated value that identifies the authentication methodology. Supported values are as follows:

0x01 ASCII — simple login, Oracle Enterprise Session Border Controller prompts for username and password

0x02 PAP — as specified in RFC 1334

0x03 CHAP — as specified in RFC 1994

service

This 8-bit field contains an enumerated value that identifies the service requesting the authentication. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

user_len

This 8-bit field contains the length of the user field in octets.

port_len

This 8-bit field contains the length of the port field in octets. As the port field is not used in the current TACACS+ authentication implementation, the port_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

rem_addr_len

This 8-bit field contains the length of the rem_addr field in octets. As the rem_addr field is not used in the current TACACS+ authentication implementation, the rem_addr_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 8-bit field contains the length of the data field in octets.

user

This variable length field contains the login name of the user to be authenticated.

port

This variable length field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

rem_addr

This variable length field contains the location of the user to be authenticated. This field contains the localhost address.

data

This optional variable length field contains miscellaneous data.

Authentication REPLY Packet

The TACACS+ daemon sends an authentication REPLY packet to the Oracle Enterprise Session Border Controller in response to a authentication START or authentication CONTINUE packet. Depending on the contents of the status field, the authentication REPLY packet either ends the authentication transaction, or continues the transaction by requesting addition information needed by the authenticator.

The authentication REPLY packet format is as follows.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x1        |
+--------+--------+--------+-------+
|     (type field contains 0x1)    |
+--------+--------+----------------+
| status |  flags |  server_msg_len |
|--------+--------+--------+-------+
|     data_len    |  server_msg ... |
+-----------------+-----------------+
|              data ...            |
+----------------------------------+
```

status

This 16-bit field contains an enumerated value that specifies the current state of the authentication process. Supported values are as follows:

0x01 PASS — the user is authenticated, thus ending the session

0x02 FAIL — the user is rejected, thus ending the session

0x04 GETUSER — daemon request for the user name

0x05 GETPASS — daemon request for the user password

0x06 RESTART — restarts the transaction, possibly because the sequence number has wrapped, or possibly because the requested authentication type is not supported by the daemon

0x07 ERROR — reports an unrecoverable error

flags

This 8-bit field contains various flags that are not under user control.

server_msg_len

This 16-bit field contains the length of the server_msg field in octets. As the server_msg field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the server_msg_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the data_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

server_msg

This optional variable length field contains a server message intended for display to the user. The current TACACS+ authentication implementation does not use this field.

data

This optional variable length field contains data pertinent to the authentication process. The current TACACS+ authentication implementation does not use this field.

Authentication CONTINUE Packet

The Oracle Enterprise Session Border Controller, acting as a TACACS+ client, sends an authentication CONTINUE packet to the TACACS+ daemon in response to a REPLY message which requested additional data required by the authenticator.

The authentication CONTINUE packet format is as follows.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x1        |
+--------+--------+----------------+
|   user_msg_len  |     data_len   |
|--------+--------+----------------+
|  flags |       user_msg ...      |
+--------+-------------------------+
|              data ...            |
+----------------------------------+
```

user_msg_len

This 16-bit field contains the length of the user_msg field in octets.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in the current TACACS+ authentication implementation, the data field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

flags

This 8-bit field contains various flags that are not under user control.

user_msg

This variable length field contains a string that responds to an information request contained in a REPLY message.

data

This optional variable length field contains miscellaneous data, often in response to a daemon request. The current TACACS+ authentication implementation does not use the data field in Authentication CONTINUE packets.

Authentication Scenarios

Each of the supported user authentication scenarios is described in terms of packet flow in the following sections.

ASCII Authentication

The Oracle Enterprise Session Border Controller initiates the authentication with an authentication START packet.

```
+----------------------------------+
|           Common Header          |
|     minor_version contains 0x0   |
|         type contains 0x1        |
+--------+--------+--------+-------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
```

```
|    0   |    N   |    N   |    0   |
+--------+--------+--------+--------+
|                port                |
|               tty10                |
+------------------------------------+
|              rem_addr              |
|         localhost address          |
+------------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len and data_len fields contain a value of 0 , as required by the TACACS+ protocol.
- The port_len and rem_addr_len fields contain the length, in octets, of the port and rem_addr fields.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.

The TCACS+ daemon returns an authentication REPLY requesting the username.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x0     |
|         type contains 0x1          |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|  0x04  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies a daemon request — 0x04 for TAC_PLUS_AUTH_STATUS_GETUSER (get username).
- The server_msg_len data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Session Border Controllerresponds with an authentication CONTINUE packet.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x0     |
|         type contains 0x1          |
+-----------------+------------------+
|   user_msg_len  |     data_len     |
|                 |        0         |
|--------+--------+------------------+
|  flags |       user_msg ...        |
+--------+---------------------------+
```

- The user_msg_len field contains the length, in octets, of the user_msg field.
- The data_len field contains a value of 0 , as required by the TACACS+ protocol.
- The user_msg field contains the username to be authenticated.

The TCACS+ daemon returns a second authentication REPLY requesting the user password.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x0     |
|         type contains 0x1          |
```

```
+--------+--------+--------+--------+
| status |  flags |  server_msg_len |
|  0x05  |        |        0        |
|--------+--------+--------+--------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies a daemon request — 0x05 for TAC_PLUS_AUTH_STATUS_GETPASS (get user password).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Session Border Controller responds with a second authentication CONTINUE packet.

```
+----------------------------------+
|           Common Header          |
|     minor_version contains 0x0   |
|         type contains 0x1        |
+-----------------+----------------+
|   user_msg_len  |     data_len   |
|                 |        0       |
|--------+--------+--------+-------+
|  flags |       user_msg ...      |
+--------+-------------------------+
```

- The user_msg_len field contains the length, in octets, of the user_msg field.
- The data_len field contains a value of 0 , as required by the TACACS+ protocol.
- The user_msg field contains the user password to be authenticated.
- Other, optional fields are not used.

The TCACS+ daemon returns a third authentication REPLY reporting the authentication result, and terminating the authentication session.

```
+----------------------------------+
|           Common Header          |
|     minor_version contains 0x0   |
|         type contains 0x1        |
+--------+--------+----------------+
| status |  flags |  server_msg_len |
|  0x01  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len , and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

PAP Authentication

The Oracle Enterprise Session Border Controller initiates the authentication with an authentication START packet.

```
+----------------------------------+
|           Common Header          |
|     minor_version contains 0x1   |
|         type contains 0x1        |
+--------+--------+--------+-------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x02  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
```

```
|    N   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|                 user               |
+------------------------------------+
|                 port               |
|                tty10               |
+------------------------------------+
|               rem_addr             |
|          localhost address         |
+------------------------------------+
|               data ...             |
+------------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x02 for TAC_PLUS_AUTHEN_TYPE_PAP (PAP login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The data_len field contains the length, in octets, of the date field.
- The user field contains the username to be authenticated.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The data field contains the password to be authenticated.

The TCACS+ daemon returns an authentication REPLY reporting the authentication result.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x1     |
|          type contains 0x1         |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|  0x01  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

CHAP Authentication

The Oracle Enterprise Session Border Controller initiates the authentication with an authentication START packet.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x1     |
|          type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x03  |  0x01  |
```

```
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
|    N   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|                 user               |
+------------------------------------+
|                 port               |
|                tty10               |
+------------------------------------+
|               rem_addr             |
|          localhost address         |
+------------------------------------+
|               data ...             |
+------------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x03 for TAC_PLUS_AUTHEN_TYPE_CHAP (CHAP login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The data_len field contains the length, in octets, of the date field.
- The user field contains the username to be authenticated.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The data field contains the password to be authenticated.

The TCACS+ daemon returns an authentication REPLY reporting the authentication result.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x1     |
|          type contains 0x1         |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|  0x01  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

TACACS+ Authorization

The Oracle Enterprise Session Border Controller uses TACACS+ services to provide administrative authorization. With TACACS+ authorization enabled, each individual ACLI command issued by an admin user is authorized by the TACACS+ authorization service. The Oracle Enterprise Session Border Controller replicates each ACLI command in its entirety, sends the command string to the authorization service, and suspends command execution until it receives an authorization response. If TACACS+ grants authorization, the pending command is executed; if authorization is

not granted, the Oracle Enterprise Session Border Controller does not execute the ACLI command, and displays an appropriate error message.

The daemon's authorization decisions are based on a database lookup. Data base records use regular expressions to associate specific command string with specific users. The construction of such records is beyond the scope of this document.

Authorization Message Exchange

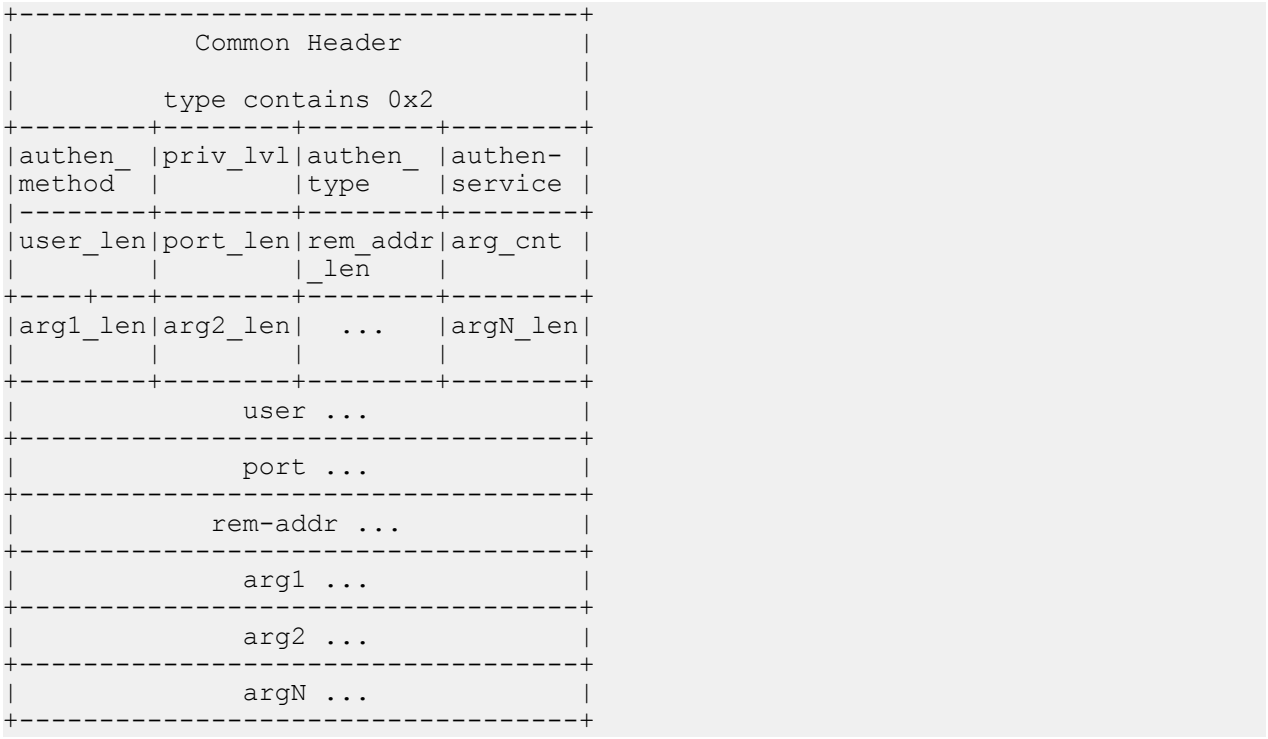
All TACACS+ authorization packets consist of a common header and a message body. Authorization packets are of two types: REQUEST and RESPONSE.

The REQUEST packet, which initiates an authorization session, is always sent by the Oracle Enterprise Session Border Controller. Upon receipt of every REQUEST, the daemon must answer with a RESPONSE packet. In the current TACACS+ implementation, the RESPONSE packet must contain an authorization decision (pass or fail). The exchange of a single REQUEST and the corresponding RESPONSE completes the authorization session.

Authorization REQUEST Packet

The Oracle Enterprise Session Border Controller, acting as a TACACS+ client, sends an authorization REQUEST packet to the TACACS+ daemon to initiate an authorization session.

The authorization REQUEST packet format is as follows.



authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the authorization subject — that is, an admin user. Because the admin user was authenticated locally by the Oracle Enterprise Session Border Controller, this field always contains a value of 0x05 , indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the authorization subject. For the current TACACS+ authorization implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the authorization subject. Because the admin user was authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the REQUEST. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the user to be authorized.

port

This variable length field contains the name of the Oracle Enterprise Session Border Controller port on which authorization is taking place. Following Cisco Systems convention, this field contains the string tty10 .

rem_addr

This variable length contains the location of the user to be authorized. This field contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative authorization requires the use of only two TACACS+ AVPs: service and cmd .

The service AVP identifies the function to be authorized. In the case of the current implementation, the attribute value is always shell . Consequently the attribute takes the follow format:

service=shell

The cmd AVP identifies the specific ACLI command to be authorized. The command is passed in its entirety, from the administrative configuration root, configure terminal, through the final command argument. For example,

cmd=configure terminal security authentication type tacacsplus

Note the equal sign (=) used in the attribute examples, indicating that both are mandatory arguments.

Authorization RESPONSE Packet

The TACACS+ daemon sends an authorization RESPONSE packet to the Oracle Enterprise Session Border Controller to report authorization results.

The authorization RESPONSE packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+-----------------+
|status  |arg_cnt | server_msg len  |
|        |        |                 |
|--------+--------+--------+--------+
|     data_len    |arg1_len|arg2_len|
|                 |        |        |
+--------+--------+--------+--------+
|   ...  |argN_len|    server_msg    |
|        |        |                  |
+--------+--------+------------------+
|               data ...             |
+-----------------------------------+
|               arg1 ...             |
+-----------------------------------+
|               arg2 ...             |
+-----------------------------------+
|               argN ...             |
+-----------------------------------+
```

status

This 8-bit field contains an enumerated value that specifies the results of the authorization process. Supported values are 0x01 (Pass), 0x10 (Fail), and 0x11 (Error). Fail indicates that the authorization service rejected the proposed operation, while Error indicates the authorization service failed

If authorization succeeds (status=0x01), the ACLI command is executed; if authorization fails, for whatever the reason (status=0x10 or 0x11), the ACLI command is not executed, and an appropriate error message is generated.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the RESPONSE. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

server_msg_len

This 16-bit field contains an integer that specifies the length, in octets, of the server_msg field.

data_len

This 16-bit field contains an integer that specifies the length, in octets, of the data field.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

server-msg

This optional variable length field contains a string that can be presented to the user.

data

This optional variable length field contains a string that can be presented to an administrative display, console, or log.

arg...

This optional variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

No arguments are generated in RESPONSE packets within the current TACACS+ implementation.

Authorization Scenarios

Successful and failed administrative authorization is described in terms of packet flow in the following sections.

Authorization Pass

The Oracle Enterprise Session Border Controller initiates the authorization with an authorization REQUEST packet.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x2        |
+--------+--------+--------+-------+
|authen_ |priv_lvl|authen_ |authen_ |
|method  |        |type    |service |
|  0x05  |  0x00  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|arg_cnt |
|        |        |_len    |        |
|    N   |    N   |    N   |    2   |
+--------+--------+--------+--------+
|arg1_len|arg2_len|      user ...    |
|        |        |                  |
|    N   |    N   |    login name    |
+--------+--------+------------------+
|               port                |
|              tty10                |
+-----------------------------------+
|             rem_addr              |
|         localhost address         |
+-----------------------------------+
|               arg1                |
|               AVP                 |
|           service=shell           |
+-----------------------------------+
|               arg2                |
|               AVP                 |
|  cmd=configure terminal security  |
+-----------------------------------+
```

- The authen_method field specifies the method used to authenticate the subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the service AVP.
- The arg2_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory service AVP.

- The arg2 field contains the mandatory cmd AVP.

The TCACS+ daemon returns a authorization RESPONSE reporting the status, and terminating the authorization session.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x2        |
+--------+--------+----------------+
| status |arg_cnt |  server_msg_len |
|  0x01  |   0    |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authorization status — 0x01 for TAC_PLUS_AUTHOR_STATUS_PASS_ADD (authorization approved).
- The arg_cnt field contains a value of 0 — the authorization RESPONSE returns no arguments.
- The server_msg_len and data_len fields both contain a value of 0, as required by the TACACS+ protocol.

Authorization Fail

The Oracle Enterprise Session Border Controller initiates the authorization with an authorization REQUEST packet.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x2        |
+--------+--------+--------+-------+
|authen_ |priv_lvl|authen_ |authen_ |
|method  |        |type    |service |
|  0x05  |  0x00  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|arg_cnt |
|        |        |_len    |        |
|    N   |    N   |    N   |    2   |
+--------+--------+--------+--------+
|arg1_len|arg2_len|      user ...    |
|        |        |                  |
|    N   |    N   |    login name    |
+--------+--------+------------------+
|                port                |
|               tty10                |
+----------------------------------+
|              rem_addr              |
|         localhost address          |
+----------------------------------+
|                arg1                |
|                AVP                 |
|           service=shell            |
+----------------------------------+
|                arg2                |
|                AVP                 |
|   cmd=configure terminal scurity   |
+----------------------------------+
```

- The authen_method field specifies the method used to authenticate the administrative subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).

- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem-addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the service AVP.
- The arg2_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory service AVP.
- The arg2 field contains the mandatory cmd AVP.

The TCACS+ daemon returns an authorization RESPONSE reporting the status, and terminating the authorization session.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+--------+--------+
| status |arg_cnt |  server_msg_len |
|  0x10  |   0    |        0        |
|--------+--------+--------+--------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authorization status — 0x10 for TAC_PLUS_AUTHOR_STATUS_FAIL (authorization rejected).
- The arg_cnt field contains a value of 0 — the authorization RESPONSE returns no arguments.
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

TACACS+ Accounting

The Oracle Enterprise Session Border Controller uses TACACS+ accounting to log administrative actions. With accounting enabled, each individual ACLI command executed by an admin user is logged by the accounting service.

Accounting Message Exchange

All TACACS+ accounting packets consist of a common header and a message body. Accounting packets are of two types: REQUEST and REPLY.

The REQUEST packet has three variant forms. The START variant initiates an accounting session; the STOP variant terminates an accounting session; the WATCHDOG variant updates the current accounting session. REQUEST packets are always sent by the Oracle Enterprise Session Border Controller. Upon receipt of every REQUEST, the daemon must answer with a REPLY packet.

A TACACS+ accounting session proceeds as follows.

1. Immediately following successful authorization of an admin user, the Oracle Enterprise Session Border Controller sends an accounting REQUEST START packet.
2. The daemon responds with an accounting REPLY packet, indicating that accounting has started.
3. For each ACLI command executed by an admin user, the Oracle Enterprise Session Border Controller sends an accounting REQUEST WATCHDOG packet requesting accounting of the ACLI command. As the Oracle Enterprise Session Border Controller sends the WATCHDOG only after an admin user’s access to the ACLI

command is authorized, the accounting function records only those commands executed by the user, not those commands for which authorization was not granted.

4. The daemon responds with an accounting REPLY packet, indicating that the ACLI operation has been recorded by the accounting function.
5. Steps 3 and 4 are repeated for each authorized ACLI operation.
6. Immediately following logout (or timeout) of an admin user, the Oracle Enterprise Session Border Controller sends an accounting REQUEST STOP packet.
7. The daemon responds with an accounting REPLY packet, indicating that accounting has stopped.

Accounting REQUEST Packet

The Oracle Enterprise Session Border Controller, acting as a TACACS+ client, sends an accounting REQUEST START variant to the TACACS+ daemon following the successful authorization of an admin user. It sends an accounting REQUEST WATCHDOG variant to the daemon following the authorization of an admin user's access to an ACLI command. It sends an accounting REQUEST STOP variant to the daemon at the conclusion of the ACLI session.

The accounting REQUEST packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|argN_len|
|        |        |        |        |
+--------+--------+--------+--------+
|argN_len|         user ...          |
+--------+---------------------------+
|              port ...             |
+-----------------------------------+
|            rem-addr ...           |
+-----------------------------------+
|              arg1 ...             |
+-----------------------------------+
|              arg2 ...             |
+-----------------------------------+
|              argN ...             |
+-----------------------------------+
```

flags

This 8-bit field contains an enumerated value that identifies the accounting REQUEST variant.

0x2 — START

0x4 — STOP

0x8 — WATCHDOG

authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the accounting subject — that is, an admin user. Because an admin user is authenticated locally by the Oracle Enterprise Session Border Controller, this field always contains a value of 0x05 , indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the accounting subject. For the current TACACS+ accounting implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the accounting subject. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the accounting REQUEST.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the accounting subject.

port

This variable length field contains the name of the Oracle Enterprise Session Border Controller port on accounting is taking place. Following Cisco System convention, this field always contains the string tty10 .

rem_addr

This variable length contains the location of the authorization subject. This field always contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative accounting requires the use of five TACACS+ AVPs: service, task-id, start_time, and stop_time.

The task_id AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, correlates session initiation, watchdog updates, and termination packets; each associated START, STOP, and WATCHDOG packet must contain matching task-id AVPs.

task_id=13578642

The start_time AVP, included in accounting REQUEST START and WATCHDOG variants, specifies the time at which a specific accounting request was initiated. The start time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

start_time=1286790650

The stop_time AVP, included in accounting REQUEST STOP variants, specifies the time at which a specific accounting session was terminated. The stop time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

stop_time=1286794250

The service AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, identifies the function subject to accounting. In the case of the current implementation, the attribute value is always shell . Consequently the attribute takes the follow format:

service=shell

The cmd AVP, included in accounting REQUEST WATCHDOG variants, identifies the specific ACLI command to be processed by the accounting service. The command is passed in its entirety, from the administrative configuration root, configure terminal, through the final command argument. For example,

cmd=configure terminal security authentication type tacacsplus

Note the equal sign (=) used in the attribute examples, indicating that all are mandatory arguments.

Accounting REPLY Packet

The TACACS+ daemon sends an accounting REPLY packet to the Oracle Enterprise Session Border Controller to report accounting results.

The accounting REPLY packet format is as follows.

server_msg_len

This 16-bit field contains the length, in octets, of the server_msg field.

data_len

This 16-bit field contains the length, in octets, of the data field.

status

This 8-bit field contains the status of the previous accounting request. Supported values are:

0x1 — Success

0x2 — Error/Failure

server_msg

This optional variable length field can contain a message intended for display to the user. This field is unused in the current TACACS+ implementation.

data

This optional variable length field can contain miscellaneous data. This field is unused in the current TACACS+ implementation.

Accounting Scenario

The Oracle Enterprise Session Border Controller initiates the accounting session with an accounting REQUEST START.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x3        |
+--------+--------+--------+-------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x02  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
|  0X01  |    N   |    N   |    N   |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|arg3_len|
|    3   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|               user               |
|    login name of an admin user   |
+----------------------------------+
|               port               |
|              tty10               |
+----------------------------------+
|             rem_addr             |
|         localhost address        |
+----------------------------------+
|               AVP                |
|          task-id=13578642        |
+----------------------------------+
|               AVP                |
|       start_time=1286790650      |
+----------------------------------+
|               AVP                |
|           service=shell          |
+----------------------------------+
```

- The flags field contains an enumerated value (0x02) that identifies an accounting REQUEST START.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.

- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TCACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has started.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x3        |
+-----------------+----------------+
|  server_msg_len |     data_len   |
|        0        |        0       |
|--------+--------+----------------+
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

The Oracle Enterprise Session Border Controller reports ACLI command execution with an accounting REQUEST WATCHDOG.

```
+----------------------------------+
|           Common Header          |
|                                  |
|         type contains 0x3        |
+--------+--------+--------+-------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x08  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
|  0X01  |    N   |    N   |    N   |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|arg3_len|
|    4   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|arg4_len|           user           |
|        | login name of admin user |
+--------+--------------------------+
|                port              |
|               tty10              |
+----------------------------------+
|              rem_addr            |
|         localhost address        |
+----------------------------------+
|                AVP               |
|          task-id=13578642        |
+----------------------------------+
|                AVP               |
|       start_time=1286790650      |
+----------------------------------+
|                AVP               |
|           service=shell          |
+----------------------------------+
|                AVP               |
```

```
|  cmd=configure terminal security  |
+-----------------------------------+
```

- The flags field contains an enumerated value (0x08) that identifies an accounting REQUEST WATCHDOG.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The arg4_len field contains the length, in octets, of the cmd AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.
- The arg4 field contains the mandatory cmd AVP.

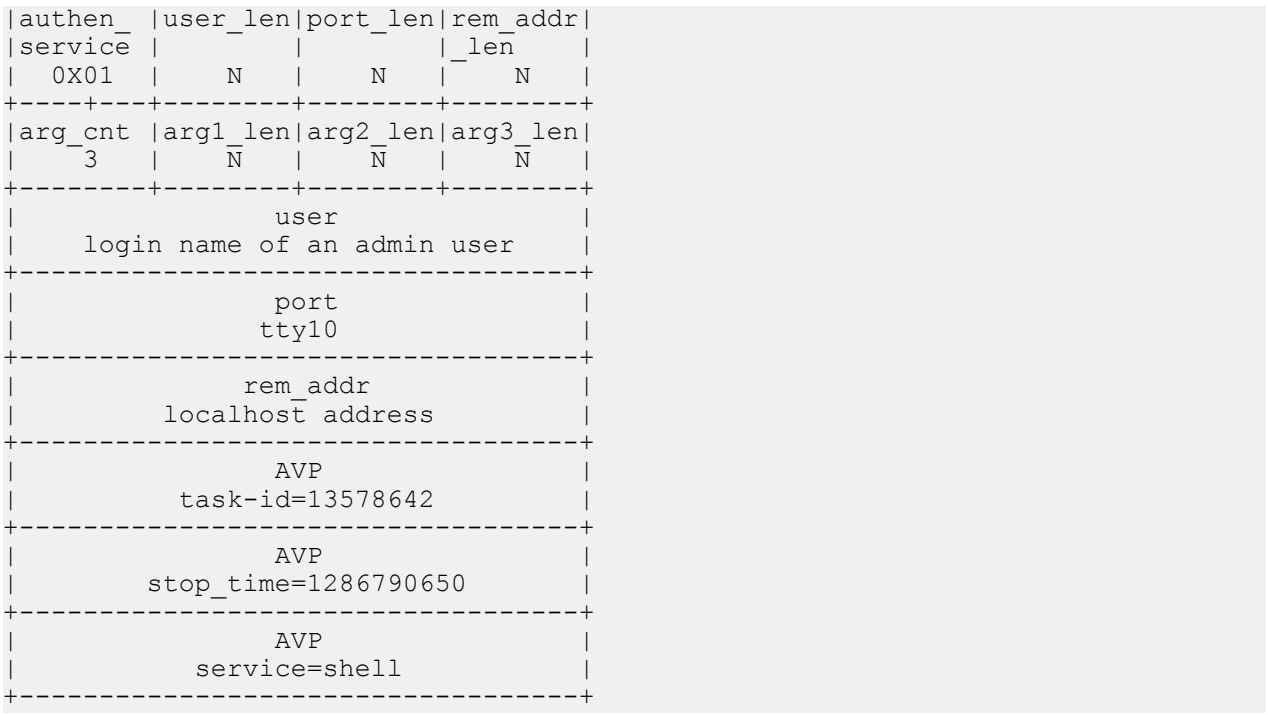
The TCACS+ daemon returns an accounting REPLY reporting the status, indicating that the ACLI operation has been processed.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+-----------------+-----------------+
|  server_msg_len |     data_len    |
|        0        |        0        |
|--------+--------+-----------------+
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

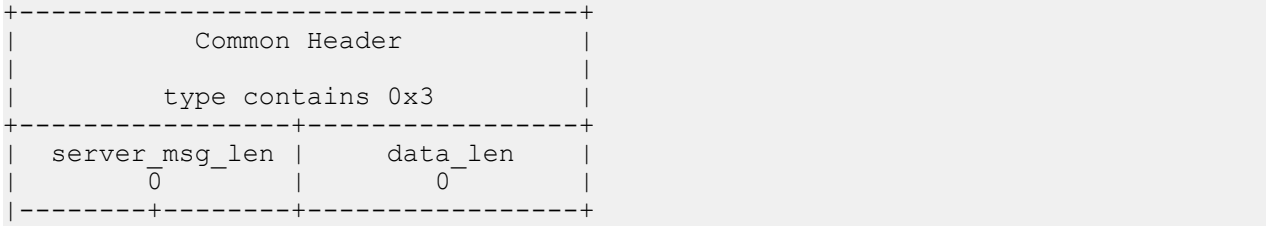
The Oracle Enterprise Session Border Controller reports an admin user logout or timeout with an accounting REQUEST STOP.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x04  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
```



- The flags field contains an enumerated value (0x04) that identifies an accounting REQUEST STOP.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Session Border Controller port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TCACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has terminated.



```
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

TACACS+ Configuration

Configuration of TACACS+ consists of the following steps.

1. Enable TACACS+ client services
2. Specify one or more TACACS+ servers (daemons)

Note: TACACS servers must be reachable from the Oracle Enterprise Session Border Controller's media interfaces. Communications to and from TACACS servers can not be from the management interfaces.

Enable TACACS+ Client Services

Use the following procedure to enable specific TACACS+ client AAA services.

1. Access the **authentication** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# authentication
ACMEPACKET(authentication)#
```

2. Use the required type attribute to specify the authentication protocol.

   Supported values are diameter, local (the default, indicating that authentication determinations are referred to a local database), radius, and tacacs.

   Specify tacacs to enable the TACACS+ AAA protocol.

```
ACMEPACKET(authentication)# type tacacs
ACMEPACKET(authentication)#
```

3. Use tacacs-authorization to enable or disable command-based authorization of admin users. Assuming type is set to tacacs , by default, authorization is enabled.

```
ACMEPACKET(authentication)# tacacs-authorization enabled
ACMEPACKET(authentication)#
```

4. Use tacacs-accounting to enable or disable accounting of admin ACLI operations. Assuming type is set to tacacs , by default, accounting is enabled.

```
ACMEPACKET(authentication)# tacacs-accounting enabled
ACMEPACKET(authentication)#
```

5. Use the server-assigned-privilege attribute to enable a proprietary TACACS+ variant that, after successful user authentication, adds an additional TACACS+ request/reply exchange. During the exchange, the Security Gateway requests the privilege level of the newly authenticated user. In response, the TACACS+ daemon returns the assigned privilege level, either user or admin . user accounts are denied access to the enable command, thus barring them from configuration level commands.

   By default, this attribute is disabled — meaning that no privilege level information is exchanged.

   Set this attribute to enabled to initiate the proprietary variant behavior.

```
ACMEPACKET(authentication)# server-assigned-privilege enabled
ACMEPACKET(authentication)#
```

6. Use the management-strategy attribute to identify the selection algorithm used to choose among multiple available TACACS+ daemons.

   Retain the default value (hunt) if only a single daemon is available.

Available algorithms are hunt and roundrobin .

- hunt — for the first transaction the Security Gateway selects the initially configured TACACS+ daemon. As long as that daemon is online and operational, the Security Gateway directs all AAA transactions to it. Otherwise, the Security Gateway selects the second-configured daemon. If the first and second daemons are offline or non-operational, the next-configured daemon is selected, and so on through the group of available daemons.
- roundrobin — for the first transaction the Security Gateway selects the initially configured TACACS+ daemon. After completing the first transaction, selects each daemon in order of configuration — in theory, evenly distributing AAA transactions to each daemon over time.

```
ACMEPACKET(authentication)# management-strategy roundrobin
ACMEPACKET(authentication)#
```

7. Type **done** to save your configuration.

Specify TACACS+ Servers

Use the following procedure to specify one or more TACACS+ servers (daemons).

1. Access the **tacacs-servers**configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# authentication
ACMEPACKET(authentication)# tacacs-servers
ACMEPACKET(tacacs-servers)#
```

2. Use the address attribute to specify the IP address of this TACACS+ daemon.

```
ACMEPACKET(tacacs-servers)# address 172.30.0.6
ACMEPACKET(tacacs-servers)#
```

Note: This address must be reachable from a media interface. The TACACs server should not be on the management interface's network.

3. Use the port attribute to identify the daemon port that receives TACACS+ client requests.

   Provide a port number within the range 1025 through 65535, or retain the default value, 49, the well-known TACACS+ port.

```
ACMEPACKET(tacacs-servers)# port 49
ACMEPACKET(tacacs-servers)#
```

4. Use the state attribute to specify the availability of this TACACS+ daemon.

   Select enabled (the default) or disabled.

   Only TACACS+ daemons that are in the enabled state are considered when running the server-selection algorithm.

```
ACMEPACKET(tacacs-servers)# state enabled
ACMEPACKET(tacacs-servers)#
```

5. Use the realm-id attribute to identify the realm that provides access to this TACACS+ deamon.

```
ACMEPACKET(tacacs-servers)# realm-id accounting
ACMEPACKET(tacacs-servers)#
```

Note: This realm must be reachable from a media interface.

6. Retain the default value for the authentication-methods attribute to specify support for all TACACS+ authentication methods (pap, chap, and ascii).

   - ascii — simple login, the Session Director prompts user for username and password
   - pap — similar to ascii method, but username and password are encapsulated in a PAP header
   - chap — authentication based on a shared-secret, which is not passed during the authentication process

```
ACMEPACKET(tacacs-servers)# authentication-methods all
ACMEPACKET(tacacs-servers)#
```

7. Use the secret attribute to provide the shared-secret used by the TACACS+ client and the daemon to encrypt and decrypt TACACS+ messages. The identical shared-secret must be configured on associated TACACS+ clients and daemons.

   Enter a 16-digit string, and ensure that the identical value is configured on the TACACS+ daemon.

   ```
   ACMEPACKET(tacacs-servers)# secret 1982100754609236
   ACMEPACKET(tacacs-servers)#
   ```

8. Use the dead-time attribute to specify, in seconds, the quarantine period imposed upon TACACS+ daemons that become unreachable. Quarrantined servers are not eligible to participate in the server-selection algorithm.

   Supported values are integers within the range 10 through 10000 seconds, with a default value of 10 .

   ```
   ACMEPACKET(tacacs-servers)# dead-interval 120
   ACMEPACKET(tacacs-servers)#
   ```

9. Type **done** to save your configuration.
10. Repeat Steps 1 through 10 to configure additional TACACS+ daemons.

    Note: After configuring TACACS+ daemons, complete TACACS+ configuration by compiling a list of available deamons.

11. From superuser mode, use the following command sequence to access authentication configuration mode.

    ```
    ACMEPACKET# configure terminal
    ACMEPACKET(configure)# security
    ACMEPACKET(security)# authentication
    ACMEPACKET(authentication)#
    ```

12. Use the management-servers attribute to identify one or more TACACS+ daemons available to provide AAA services.

    Daemons are identified by IP address and must have been previously configured as described above.

    The following example identifies three available TACACS+ daemons. Note that the list is delimited by left and right parentheses, and list items are separated by space characters.

    ```
    ACMEPACKET(tacacs-servers)# management-servers (172.30.0.6 172.30.1.8
    172.30.2.10)
    ACMEPACKET(tacacs-servers)#
    ```

    The following example deletes the current list.

    ```
    ACMEPACKET(tacacs-servers)# management-servers ()
    ACMEPACKET(tacacs-servers)#
    ```

Managing TACACS+ Operations

TACACS+ management is supported by the following utilities.

TACACS+ MIB

An Oracle proprietary MIB provides external access to TACACS+ statistics.

MIB counters are contained in the apSecurityTacacsPlusStatsTable that is defined as follows.

```
SEQUENCE {
    apSecurityTacacsPlusCliCommands                Counter32
    apSecurityTacacsPlusSuccess Authentications    Counter32
    apSecurityTacacsPlusFailureAuthentications     Counter32
    apSecurityTacacsPlusSuccess Authorizations     Counter32
    apSecurityTacacsPlusFailureAuthorizations      Counter32
}
```

apSecuritysTacacsPlusStats Table (1.3.6.1.4.1.9148.3.9.9.4)		
Object Name	**Object OID**	**Description**
apSecurityTacacsCliCommands	1.3.6.1.4.1.9148.3.9.1.4.3	Global counter for ACLI commands sent to TACACS+ Accounting
apSecurityTacacsSuccess Authentications	1.3.6.1.4.1.9148.3.9.1.4.4	Global counter for the number of successful TACACS+ authentications
apSecurityTacacsFailureAuthentications	1.3.6.1.4.1.9148.3.9.1.4.5	Global counter for the number of unsuccessful TACACS+ authentications
apSecurityTacacsSuccess Authorizations	1.3.6.1.4.1.9148.3.9.1.4.6	Global counter for the number of successful TACACS+ authorizations
apSecurityTacacsFailure Authorizations	1.3.6.1.4.1.9148.3.9.1.4.7	Global counter for the number of unsuccessful TACACS+ authorizations

SNMP Trap

SNMP traps are issued when

- a TACACS+ daemon becomes unreachable
- an unreachable TACACS+ daemon becomes reachable
- an authentication error occurs
- an authorization error occurs

ACLI show Command

The show tacacs stats command displays the following statistics.

- number of ACLI commands sent for TAGACS+ accounting
- number of successful TACACS+ authentications
- number of failed TACACS+ authentications
- number of successful TACACS+ authorizations
- number of failed TACACS+ authentications
- the IP address of the TACACS+ daemon used for the last transaction

TACACS+ Logging

All messages between the Oracle Enterprise Session Border Controller and the TACACS+ daemon are logged in a cleartext format, allowing an admin user to view all data exchange, except for password information.

H

RADIUS Authentication

A security feature that extends beyond the designation of ACLI User and Superuser privileges, the User Authentication and Access control feature supports authentication using your RADIUS server(s). In addition, you can set two levels of privilege, one for all privileges and more limited set that is read-only.

User authentication configuration also allows you to use local authentication, localizing security to the Oracle Enterprise Session Border Controller ACLI log-in modes. These modes are User and Superuser, each requiring a separate password.

The components involved in the RADIUS-based user authentication architecture are the Oracle Enterprise Session Border Controller and your RADIUS server(s). In these roles:

- The Oracle Enterprise Session Border Controller restricts access and requires authentication via the RADIUS server; the Oracle Enterprise Session Border Controller communicates with the RADIUS server using either port 1812 or 1645, but does not know if the RADIUS server listens on these ports
- Your RADIUS server provides an alternative method for defining Oracle Enterprise Session Border Controller users and authenticating them via RADIUS; the RADIUS server supports the VSA called ACME_USER_CLASS, which specifies what kind of user is requesting authentication and what privileges should be granted.

  The Oracle Enterprise Session Border Controller also supports the use of the Cisco Systems Inc.™ Cisco-AVPair vendor specific attribute (VSA). This attribute allows for successful administrator login to servers that do not support the Oracle authorization VSA. While using RADIUS-based authentication, the Oracle Enterprise Session Border Controller authorizes you to enter Superuser mode locally even when your RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. For this VSA, the Vendor-ID is 1 and the Vendor-Type is 9. The list below shows the values this attribute can return, and the result of each:

  - shell:priv-lvl=15—User automatically logged in as an administrator
  - shell:priv-lvl=1—User logged in at the user level, and not allowed to become an administrator
  - Any other value—User rejected

When RADIUS user authentication is enabled, the Oracle Enterprise Session Border Controller communicates with one or more configured RADIUS servers that validates the user and specifies privileges. On the Oracle Enterprise Session Border Controller, you configure:

- What type of authentication you want to use on the Oracle Enterprise Session Border Controller
- If you are using RADIUS authentication, you set the port from which you want the Oracle Enterprise Session Border Controller to send messages
- If you are using RADIUS authentication, you also set the protocol type you want the Oracle Enterprise Session Border Controller and RADIUS server to use for secure communication

Although most common set-ups use two RADIUS servers to support this feature, you are allowed to configure up to six. Among other settings for the server, there is a class parameter that specifies whether the Oracle Enterprise Session Border Controller should consider a specific server as primary or secondary. As implied by these designation, the primary servers are used first for authentication, and the secondary servers are used as backups. If you configure more than one primary and one secondary server, the Oracle Enterprise Session Border Controller will choose servers to which it sends traffic in a round-robin strategy. For example, if you specify three servers are primary, the Oracle Enterprise Session Border Controller will round-robin to select a server until it finds an appropriate one; it will do the same for secondary servers.

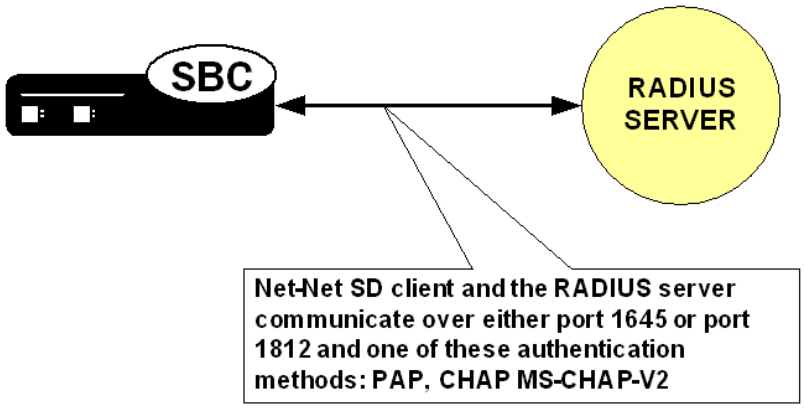
The VSA attribute assists with enforcement of access levels by containing one of the three following classes:

- None—All access denied
- User—Monitoring privileges are granted; your user prompt will resemble ACMEPACKET>
- Admin—All privileges are granted (monitoring, configuration, etc.); your user prompt will resemble ACMEPACKET#

Once it has selected a RADIUS server, the Oracle Enterprise Session Border Controller initiates communication and proceeds with the authentication process. The authentication process between the Oracle Enterprise Session Border Controller and the RADIUS server takes place uses one of three methods, all of which are defined by RFCs:

Protocol	RFC
PAP (Password Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992
CHAP (Challenge Handshake Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992 W. Simpson, PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996
MS-CHAP-V2	G. Zorn, Microsoft PPP CHAP Extensions, Version 2, RFC 2759, January 2000

Note: MS-CHAP-V2 support includes authentication only; password exchange is not supported or allowed on the Oracle Enterprise Session Border Controller.



PAP Handshake

For PAP, user credentials are sent to the RADIUS server include the user name and password attribute. The value of the User-Password attribute is calculated as specified in RFC 2865.

PAP Client Request Example

```
Radius Protocol
Code: Access Request (1)
  Packet identifier: 0x4 (4)
  Length: 61
  Authenticator: 0x0000708D00002C5900002EB600003F37
  Attribute value pairs
    t:User Name(1) l:11, value:”TESTUSER1”
      User-Name: TESTUSER1
    t:User Password (2) l:18, value:739B3A0F25094E4B3CDA18AB69EB9E4
    t:NAS IP Address(4) l:6, value:168.192.68.8
      Nas IP Address: 168.192.68.8(168.192.68.8)
    t:NAS Port(5) l:6, value:118751232
```

PAP RADIUS Response

```
Radius Protocol
  Code: Access Accept (2)
  Packet identifier: 0x4 (4)
  Length: 20
  Authenticator: 0x36BD589C1577FD11E8C3B5BB223748
```

CHAP Handshake

When the authentication mode is CHAP, the user credentials sent to the RADIUS server include “username,” “CHAP-Password,” and “CHAP-Challenge.” The “CHAP-Password” credential uses MD-5 one way. This is calculated over this series of the following values, in this order: challenge-id (which for the Oracle Enterprise Session Border Controller is always 0), followed by the user password, and then the challenge (as specified in RFC 1994, section 4.1).

CHAP Client Request Example

```
Radius Protocol
  Code: Access Request (1)
  Packet identifier: 0x5 (5)
  Length: 80
  Authenticator: 0x0000396C000079860000312A00006558
  Attribute value pairs
    t:User Name(1) l:11, value:”TESTUSER1”
      User-Name: TESTUSER1
    t:CHAP Password (3) l:19, value:003D4B1645554E881231ED7A137DD54FBF
    t:CHAP Challenge (60) l:18, value: 000396C000079860000312A00006558
    t:NAS IP Address(4) l:6, value:168.192.68.8
      Nas IP Address: 168.192.68.8(168.192.68.8)
    t:NAS Port(5) l:6, value:118751232
```

CHAP RADIUS Response

```
Radius Protocol
  Code: Access Accept (2)
  Packet identifier: 0x4 (4)
  Length: 20
  Authenticator: 0x3BE89EED1B43D91D80EB2562E9D65392
```

MS-CHAP-v2 Handshake

When the authentication method is MS-CHAP-v2, the user credentials sent to the RADIUS server in the Access-Request packet are:

- username
- MS-CHAP2-Response—Specified in RFC 2548, Microsoft vendor-specific RADIUS attributes
- MS-CHAP2-Challenge—Serves as a challenge to the RADIUS server

If the RADIUS authentication is successful, the Access-Accept packet from the RADIUS server must include an MS-CHAP2-Success attribute calculated using the MS-CHAP-Challenge attribute included in the Access-Request. The calculation of MS-CHAP2-Success must be carried out as specified in RFC 2759. The Oracle Enterprise Session Border Controller verifies that the MS-CHAP2-Success attribute matches with the calculated value. If the values do not match, the authentication is treated as a failure.

MS-CHAP-v2 Client Request Example

Some values have been abbreviated.

```
Radius Protocol
  Code: Access Request (1)
  Packet identifier: 0x5 (5)
  Length: 80
  Authenticator: 0x0000024C000046B30000339F00000B78
  Attribute value pairs
    t:User Name(1) l:11, value:”TESTUSER1”
      User-Name: TESTUSER1
    t:Vendor Specific(26) l:24, vendor:Microsoft(311)
    t:MS CHAP Challenge(11) l:18, value:0000024C000046B30000339F00000B78
    t:Vendor Specific(26) l:58, vendor:Microsoft(311)
    t:MS CHAP2 Response(25) l:52, value:
00000000024C000046B30000339F00000B78...
    t:NAS IP Address(4) l:6, value:168.192.68.8
      Nas IP Address: 168.192.68.8(168.192.68.8)
    t:NAS Port(5) l:6, value:118751232
```

MS-CHAP-v2 RADIUS Response

```
Radius Protocol
  Code: Access Accept (2)
  Packet identifier: 0x6 (6)
  Length: 179
  Authenticator: 0xECB4E59515AD64A2D21FC6D5F14D0CC0
  Attribute value pairs
    t:Vendor Specific(26) l:51, vendor:Microsoft(311)
      t:MS CHAP Success(11) l:45, value:003533s33d38454435324431354538463l3...
    t:Vendor Specific(26) l:42, vendor:Microsoft(311)
      t:MS MPPE Recv Key(17) l:36, value:96C6325D22513CED178F770093F149CBBA...
    t:Vendor Specific(26) l:42, vendor:Microsoft(311)
      t:MS MPPE Send Key(16) l:36, value:9EC9316DBFA701FF0499D36A1032678143...
    t:Vendor Specific(26) l:12, vendor:Microsoft(311)
      t:MS MPPE Encryption Policy(7) l:6, value:00000001
    t:Vendor Specific(26) l:12, vendor:Microsoft(311)
      t:MS MPPE Encryption Type(8) l:6, value:00000006
```

Management Protocol Behavior

When you use local authentication, management protocols behave the same way that they do when you are not using RADIUS servers. When you are using RADIUS servers for authentication, management protocols behave as described in this section.

- Telnet—The “user” or admin accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication. If authentication is successful, the user is granted privileges depending on the ACME_USER_CLASS VSA attribute.
- FTP—The “user” or admin accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.

- SSH in pass-through mode—When SSH is in pass through mode, the Oracle Enterprise Session Border Controller behave the same way that it does for Telnet.
- SSH in non-pass-through mode—When you create an SSH account on the Oracle Enterprise Session Border Controller, you are asked to supply a user name and password. Once local authentication succeeds, you are prompted for the ACLI user name and password. If your user ACLI name is user, then you are authenticated locally. Otherwise, you are authenticated using the RADIUS server. If RADIUS authentication is successful, the privileges you are granted depend on the ACME_USER_CLASS VSA attribute.
- SFTP in pass-through mode—If you do not configure an SSH account on the Oracle Enterprise Session Border Controller, the RADIUS server is contacted for authentication for any user that does not have the user name user. The Oracle Enterprise Session Border Controller uses local authentication if the user name is user.
- SFTP in non-pass-through mode—The “user” or admin accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.

RADIUS Authentication Configuration

To enable RADIUS authentication and user access on your Oracle Enterprise Session Border Controller, you need to configure global parameters for the feature and then configure the RADIUS servers that you want to use.

Global Authentication Settings

To configure the global authentication settings:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type security and press Enter.

   ```
   ACMEPACKET(configure)# security
   ```

3. Type authentication and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(security)# authentication
   ACMEPACKET(authentication)#
   ```

   From here, you can view the entire menu for the authentication configuration by typing a ?. You can set global parameters for authentication. You can also configure individual RADIUS servers; instructions for configuring RADIUS server appear in the next section.

4. type—Set the type of user authentication you want to use on this Oracle Enterprise Session Border Controller. The default value is local. The valid values are:

   - local | radius

5. protocol—If you are using RADIUS user authentication, set the protocol type to use with your RADIUS server(s). The default is pap. The valid values are:

   - pap | chap | mschapv2

6. source-port—Set the number of the port you want to use from message sent from the Oracle Enterprise Session Border Controller to the RADIUS server. The default value is 1812. The valid values are:

   - 1645 | 1812

7. allow-local-authorization—Set this parameter to enabled if you want the Oracle Enterprise Session Border Controller to authorize users to enter Superuser (administrative) mode locally even when your RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. The default for this parameter is disabled.

RADIUS Server Settings

The parameters you set for individual RADIUS servers identify the RADIUS server, establish a password common to the Oracle Enterprise Session Border Controller and the server, and establish trying times.

Setting the class and the authentication methods for the RADIUS servers can determine how and when they are used in the authentication process.

To configure a RADIUS server to use for authentication:

1. Access the RADIUS server submenu from the main authentication configuration:

```
ACMEPACKET(authentication)# radius-servers
ACMEPACKET(radius-servers)#
```

2. address—Set the remote IP address for the RADIUS server. There is no default value, and you are required to configure this address.
3. port—Set the port at the remote IP address for the RADIUS server. The default port is set to 1812. The valid values are:
   - 1645 | 1812
4. state—Set the state of the RADIUS server. Enable this parameter to use this RADIUS server to authenticate users. The default value is enabled. The valid values are:
   - enabled | disabled
5. secret—Set the password that the RADIUS server and the Oracle Enterprise Session Border Controller share. This password is transmitted between the two when the request for authentication is initiated; this ensures that the RADIUS server is communicating with the correct client.
6. nas-id—Set the NAS ID for the RADIUS server. There is no default for this parameter.
7. retry-limit—Set the number of times that you want the Oracle Enterprise Session Border Controller to retry for authentication information from this RADIUS server. The default value is 3. The valid range is:
   - Minimum—1
   - Maximum—5

   If the RADIUS server does not respond within this number of tries, the Oracle Enterprise Session Border Controller marks is as dead.
8. retry-time—Set the amount of time (in seconds) that you want theOracle Enterprise Session Border Controller to wait before retrying for authentication from this RADIUS server. The default value is 5. The valid range is:
   - Minimum—5
   - Maximum—10
9. dead-time—Set the amount of time in seconds before the Oracle Enterprise Session Border Controller retries a RADIUS server that it has designated as dead because that server did not respond within the maximum number of retries. The default is 10. The valid range is:
   - Minimum—10
   - Maximum—10000
10. maximum-sessions—Set the maximum number of outstanding sessions for this RADIUS server. The default value is 255. The valid range is:
    - Minimum—1
    - Maximum—255
11. class—Set the class of this RADIUS server as either primary or secondary. A connection to the primary server is tried before a connection to the secondary server is tried. The default value is primary. Valid values are:
    - primary | secondary

    The Oracle Enterprise Session Border Controller tries to initiate contact with primary RADIUS servers first, and then tries the secondary servers if it cannot reach any of the primary ones.

    If you configure more than one RADIUS server as primary, the Oracle Enterprise Session Border Controller chooses the one with which it communicates using a round-robin strategy. The same strategy applies to the selection of secondary servers if there is more than one.
12. authentication-methods—Set the authentication method you want the Oracle Enterprise Session Border Controller to use with this RADIUS server. The default value is pap. Valid values are:

- all | pap | chap | mschapv2

  This parameter has a specific relationship to the global protocol parameter for the authentication configuration, and you should exercise care when setting it. If the authentication method that you set for the RADIUS server does not match the global authentication protocol, then the RADIUS server is not used. The Oracle Enterprise Session Border Controller simply overlooks it and does not send authentication requests to it. You can enable use of the server by changing the global authentication protocol so that it matches.

13. Save your work and activate your configuration.