Oracle® Communications Subscriber-Aware Load Balancer

Essentials Guide
Release S-CZ7.2.10

April 2017

Notices

Contents

About this guide

Version S-Cz7.2.10 provides an updated release of the Oracle Communications Subscriber-Aware Load Balancer (SLB). This guide describes that release.

This guide is written for network administrators and architects, and provides information about the SLB configuration. For information on configuration and operation of Session Boarder Controller (SBC) and Unified Session Manager (USM) SLB cluster members, refer to the related documentation set for the release used. Each of these SBC and USM releases support SLB cluster membership, in addition to the full complement of other SBC functionality.

Related Documentation

The following table describes the documentation set for this release.

Document Name	Document Description
Oracle Acme Packet 6100 Hardware Installation and Maintenance Guide	Contains information about the components, installation, and maintenance of the Acme Packet 6100.
Oracle Acme Packet 6300 Hardware Installation and Maintenance Guide	Contains information about the components, installation, and maintenance of the Acme Packet 6300.
Oracle Acme Packet 4500 Hardware Installation and Maintenance Guide	Contains information about the components, installation, and maintenance of the Acme Packet 4500.
Oracle Acme Packet 4600 Hardware Installation and Maintenance Guide	Contains information about the components, installation, and maintenance of the Acme Packet 4600.
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 SBC.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about SBC 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, numbers, and descriptions, as well as examples of scalar and table objects.
Accounting Guide	Contains information about the SBC's accounting support, including details about RADIUS accounting.

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

Revision History

Date	Description
April 2015	Initial Release of software version S-Cz7.2.10
June 2015	Adds Other Known Issues section and known issue about overlapping networks.
October 2015	Updates changes to the Known Issues list for S-Cz7.2.10p3.
February 2016	Adds Acme Packet 4600 to list of supported cluster member platforms.
March 2016	Adds OCSBC-S-Cz7.3.0 to list of supported SBC releases.
August 2016	Added that you can have only 1 SLB per cluster and each SBC must be associated with only 1 cluster.
April 2017	Removes the description of the tls-profile parameter from the tunnel-config element because it is not functional. Replaces incorrect references to product names

1

Introduction and Overview

As service providers deploy larger and larger SIP access networks, scalability problems are presenting unique challenges, particularly from an operational standpoint. Deployments that scale beyond the number of users serviceable by a single Session Boarder Controller (SBC) – as well as deployments that use a geographically redundant SBC for catastrophic fail over purposes – encounter edge reachability problems. In general there are two coarse techniques that carriers use today to support end-point populations that exceed one SBC's capacity: they will either use a DNS-based distribution mechanism, or they will pre-provision endpoint to point to specific SBCs (manually load balancing them). Each of these solutions has its drawbacks. End users – many of them familiar with load balancing equipment deployed to scale protocols such as HTTP or SMTP – have expressed interest in a device that will perform dedicated load balancing for their SIP endpoint.

The Subscriber-Aware Load Balancer (SLB) addresses the need for scaling a network edge to millions of endpoint. Designed as a standalone system (historically an Acme Packet 4500, now with this release an Acme Packet 6100 has been added) capable of supporting up to two million users for the Acme Packet 4500 and ten million endpoints for the Acme Packet 6100 (where an endpoint is defined as a unique source and destination IP address), and flexibly deployable into existing network topologies, the SLB aggregates signaling from large endpoint populations to reduce the edge reachability problem by an order of magnitude.

Functional Overview

The Subscriber-Aware Load Balancer (SLB) is a discrete network element that processes all SIP end-point signaling traffic entering the service provider network. The SLB is not necessarily the first network device to receive signaling traffic, as, depending on network topology, additional network components (for example, routers, network address translators, and so on) can lie between the end-point and the SLB.

Upon receipt of a SIP packet from an unknown source, the SLB uses a provisioned policy to select an appropriate next-hop Session Border Controller (SBC) for traffic originated by that end-point. Subsequent packets from the same end-point are forwarded to the same SBC. The first packet, the one used to make the route decision, and all subsequent packets sent through the SLB to the next-hop SBC are encapsulated within an IP-in-IP format as defined in RFC 2003, IP Encapsulation within IP.

SBCs that participate in the load balancing-enabled deployment are enhanced by several capabilities. First, the SBC supports RFC 2003 tunnel for both packet transmission and reception. Second, the SBC periodically transmits health and performance data to the SLB; such information is evaluated and entered into the SLB's route determination algorithm. Lastly, the SBC participates in any SLB-initiated rebalance operation, as described in the Rebalancing section. A group of SBCs, with the above-listed capabilities, that receive signaling traffic from the SLB, is referred to as a cluster.

The IP-in-IP encapsulation technique provides SLB transparency to the terminating SBC. That is, when an SBC receives an encapsulated packet via the SLB, it can discard the outer encapsulation leaving behind an identical packet as transmitted originally by the end-point. Visibility into the actual packet transmitted by the end-point is necessary to provide certain services in the SBC (for example, hosted NAT traversal, session-agent matching, and so on). A secondary goal achieved by using this encapsulation technique is that it provides a disassociation function between an SBC's connected network and its SIP reachability. That is, an SBC can be assigned any IP address it wants from a network topology standpoint, yet still process SIP packets as though it were logically situated elsewhere at Layer 5. In a larger sense, the physicality of the SBC is no longer important; like-configured, logically identical SBCs can be spread all over the globe.

Balancing and Rebalancing

The SLB performs two primary functions as the front-end to a Session Boarder Controller/Unified Session Manager (SBC/USM) cluster: balancing traffic and rebalancing traffic. There are several key distinctions, which are described in the following two sections.

Balancing

Balancing is defined as the distribution of new endpoints (a combination of unique source and destination IP address pairs) among the members of the cluster. The SLB balances traffic based upon its configured policies (refer to Load Balancer Policy Configuration for policy description and details), or, in the absence of configured policies, with a default round-robin procedure. Load balancer policies provide a flexible means of directing traffic to appropriate groups of SBCs. As initial packets arrive at the SLB from unknown (previously unseen) endpoints, they are passed to a resident software process that consults its policy engine to determine an appropriate destination (clustered SBC) for each endpoint. Regardless of the distribution algorithm, policy-based or round-robin, the SLB chooses an SBC from among all equally weighted candidates, giving preference to those with the lowest current occupancy rate, defined as the number of endpoints already present on that system relative to its maximum endpoint capacity.

Even though each clustered SBC regularly reports CPU data to the SLB, the SBC's CPU utilization is not factored into the preference of one SBC over another. Rather, an SBC whose CPU utilization rate, determined using a per-thread CPU load check of the busiest call-related threads (SIP and MBCD), exceeds its load limit threshold (by default, 90%) is excluded from the list of candidates. For example, assuming that both SBCs are licensed for the same number of sessions, an SBC with a CPU load of 89% and a current occupancy of 10,000 endpoint will have equal footing with an SBC with a CPU load of 10% and a current occupancy of 10,000 endpoint. But an SBC with a CPU load of 90% and an occupancy of 0 endpoint will never receive new assignments from the SLB, until its CPU utilization rate falls below the 90% threshold.

Rebalancing

Rebalancing, as opposed to balancing, is taking some number of existing endpoints from functioning SBCs and redistributing these existing endpoints between current cluster members. Rebalancing can be automatically scheduled when a new SBC joins an existing cluster, or immediately invoked with the Acme Packet Command Line Interface (ACLI). When an SBC exits a cluster, whatever the reason, all of its endpoints are invalidated on the SLB and those endpoints are essentially balanced when they revisit the SLB.

A new SBC joins an existing cluster by initiating the establishment of an IP-in-IP tunnel between itself and the SLB. During an initial handshake the SBC designates which SLB service port or ports it is prepared to support. If there are existing SBCs supporting these designated service ports, the SLB instructs some or all of these SBCs to divest themselves of a specified number of endpoints. The SLB calculates the number of divested endpoints based upon the overall occupancy of that service relative to the SLB's contribution to that occupancy. Existing cluster members not advertising support for service ports designated by the new cluster member are excluded from the rebalance queue.

The SLB sequences through eligible cluster members one at a time, using a proprietary protocol from Acme Packet to request nomination and removal of eligible endpoints. The SBC replies with a CCP response that lists candidate endpoints. The SLB removes existing forwarding rules associated with those endpoints, and repeats the CCP request/response process until the cluster member divests itself of the specified number of endpoints.

When the divested endpoints re-engage with the SLB (upon their next scheduled registration refresh, for example), the SLB lacks a forwarding rule that maps them to a specific SBC. Consequently, the message is passed up to the software processes running on the SLB's host, which chooses a new SBC destination for that endpoint – presumably, the new cluster member that has the most available capacity.

The cluster member, after being requested to nominate endpoints for rebalancing, uses several criteria for choosing the most attractive candidates. As part of its standard SIP processing performed by SBCs, the cluster member is aware of the expiry times for all of the entries in its SIP registration cache. Therefore, the cluster member can predict with a high degree of accuracy when any given endpoint will be signaling back into the cluster. As the forwarding rules on the cluster member are triggered by endpoint messages, the cluster member considers an endpoint whose registration entry is due to expire shortly an attractive candidate for rebalance. Note, however, that in many cases it is not prudent to nominate endpoints whose SIP registration cache entries are due to expire immediately, as this can cause a race condition between the CCP response and the SIP REGISTER message from the endpoint to the SIP registration function. To avoid this potential dilemma, cluster members are equipped with the ability to skip ahead to candidates whose expiry is not immediate.

Further, each cluster member categorizes the endpoints stored in its cache based upon a priority value that is determined via the SLB's distribution policy (see Distribution Policy Configuration for more details). It nominates endpoints from its lowest priority buckets first.

Finally, the SLB does not rebalance an active SIP endpoint — an endpoint engaged in a phone conversation.

After removing endpoints from the first cluster member, the SLB moves to the next cluster member in the rebalance queue and uses the same CPP request/response exchange to remove additional endpoints. The same procedure repeats for additional cluster members until the SLB attains the target number of divested endpoints.

When the divested endpoints re-engage with the SLB (upon their next scheduled registration refresh, for example), the SLB lacks a forwarding rule that maps them to a specific cluster member. Consequently, the message is passed up to the software processes running on the SLB's host, which chooses a new destination for that endpoint – presumably, the new cluster member that has the most available capacity.

IPv4 IPv6 Dual Stack

While major carriers are proceeding toward a pure IPv6 network for next generation services, current practicalities require the continued support of IPv4 handsets and other devices. As a result, the SLB provides support for single non-channelized physical interfaces that support both IPv4 and IPv6 ingress and egress on the same network interface.

Support for the dual stack interface requires no new additional configuration elements, and is provided by the proper configuration of the following elements in the ACLI Configuration Guide:

Configuration Element	Section containing configuration element in the ACLI Configuration Guide documentation
Physical Interfaces	(Platform) Physical Interfaces: SLB in the System Configuration chapter. (Release versions S-Cx6.3.x, S-Cx6.4.x, S-Cz7.1.2, or S-Cz7.2.0)

Configuration Element	Section containing configuration element in the ACLI Configuration Guide documentation
IPv4 Network Interfaces	IPv4 Address Configuration in the System Configuration chapter. (Release versions S-Cx6.3.x, S-Cx6.4.x, S-Cz7.1.2, or S-Cz7.2.0)
IPv6 Network Interfaces	Configuring Network Interfaces: SLB, Licensing, Globally Enabling IPv6, IPv6 Address Configuration, IPv6 Default Gateway, and Network Interfaces and IPv6 in the System Configuration chapter. (Release versions S-Cx6.3.x,S-Cx 6.4.x, S-Cz7.1.2, or S-Cz7.2.0)
IPv4 & IPv6 SIP Interfaces	SIP Interfaces in the SIP Signalling Services chapter. (Release versions S-Cx6.3.x, S-Cx6.4.x, S-Cz7.1.2, or S-Cz7.2.0)
IPv4 & IPv6 Realms	Realm Configuration in the Realms and Nested Realms chapter. (Release versions S-Cx6.3.x,S-Cx 6.4.x,S-Cz 7.1.2, or S-Cz 7.2.0)

SLB Cluster Member Graceful Shutdown

When it becomes necessary to temporarily remove an Oracle Communications Session Border Controller (SBC) from active service, and make it available only for administrative purposes, the user issues a **set-system-state offline** ACLI command. The SBC begins a graceful shutdown. The shutdown is graceful in that active calls and registrations are not affected, but new calls and registrations are rejected except as discussed below. When the user issues the command, the SBC goes into **becoming offline** mode. Once there are no active SIP sessions and no active SIP registrations in the system, the SBC transitions to **offline** mode. If the SBC is a member of an Oracle Communications Subscriber-Aware Load Balancer (SLB) Cluster, the offline status is communicated to the SLB when the user issues the **set-system-state offline** command, and the SLB excludes the offline SBC in future endpoint (re)balancing algorithms.

A version of this SBC graceful shutdown procedure exists in SBC releases previous to S-CZ7.3.0, but the procedure is enhanced for this and future releases. Previous versions only looked at active SIP sessions (calls), without monitoring active SIP registrations, and did not attempt to manipulate the period of time that active calls and registrations lingered on the SBC. The problem with this approach was that in the interval between setting the SBC to **offline** mode, and the subscriber registrations expiring, any inactive subscriber was essentially unreachable. With some carriers setting registration expiry timers to an hour or more (or 30 minutes in between registration refresh), this may have resulted in significant periods of unreachability. With this release, the new **sip-config** parameter **retry-after-upon-offline** is used to minimize the amount of time active calls and registrations keep the SBC from going completely offline.

The SBC side of this graceful shutdown procedure is followed with or without the SBC being a member of an SLB cluster. The graceful shutdown procedure is limited only to SIP calls and registrations.

High-level Procedure for Graceful SBC Shutdown

In its simplest form, this is the graceful shutdown procedure. Details and exceptions to this procedure when there are active calls or registrations are discussed in later paragraphs. The first six actions are performed whether or not the SBC is part of an Oracle Communications Subscriber-Aware Load Balancer (SLB) Cluster

- The SBC receives the **set-system-state offline** command.
- The SBC transitions to **becoming offline** mode.
- The SBC accepts calls and subscribes from registered endpoints.
- The SBC rejects calls from non-registered endpoints.
- The SBC rejects new registrations with a **503 Service Unavailable** error message.
- The SBC checks the number SIP INVITE based sessions and number of SIP registrations. When both counts are 0, the SBC transitions to the **offline** state.

Note: Previous versions only looked at active SIP sessions (calls), without monitoring active SIP registrations.

If the SBC is part of an SLB Cluster:

- The SLB client on the SBC changes its cluster status to **shutdown** state.
- The SBC informs the SLB that it is offline.
- The SLB ceases to forward new end-points to the SBC and puts the SBC in a shutdown state.
- SLB continues to forward all messages for existing registered endpoints to the offline SBC.
- The SBC continues to send heartbeat updates the SLB as before.

Detailed Description of Graceful Shutdowns with Active SIP Calls or Registrations

This is the procedure when active SIP calls or registrations are on an SBC.

When the system receives the **set-system-state offline** command, it transitions to **becoming offline** mode. It begins checking the number of SIP-INVITE-based sessions and the number of SIP registrations, and continues to check them when sessions complete or registrations expire while it is in **becoming offline** mode. When both counts reach zero, the system transitions to **offline mode**. If the system is a member of aOracle Communications Subscriber-Aware Load Balancer (SLB) Cluster, the SLB client on the SBC changes its cluster status to the **shutdown** state, and informs the SLB that it is **offline**. The SLB ceases to forward new end-points to the SBC and lists the SBC in a **shutdown** state on the SLB. The SBC continues to send heartbeat updates to the SLB as before.

Active calls continue normally when the SBC is in **becoming offline** mode. If SIP refresh registrations arrive for endpoints that have active calls, they are accepted. However, the expiry of these endpoints is reduced to the configurable **retry-after-upon-offline** timer value (in seconds) defined under **sip-config** on the SBC. This timer should be configured to be a much lower time interval than originally requested by the refresh registrations, so that endpoints refresh sooner and thus the registrations expire as closely as possible to when the active call ends. If the new timer value configured in **retry-after-upon-offline** is greater than the existing registration requested refresh value, or if its value is ‘0’ (unconfigured), the original registration refresh request is honored.

Refresh registrations for endpoints that do not have any active calls are rejected with a configurable response code defined in the **sip-config reg-reject-response-upon-offline** parameter. The default for this parameter is the **503 Service Unavailable** message. It includes a **Retry-After** header with a configurable timer set in **retry-after-upon-offline**. If the value of the configuration is 0 (unconfigured), the header is not included in the rejection message. Once these refreshes are rejected, SBC immediately removes such endpoints from its registration cache. It is a force remove. De-registrations are forwarded to the core. There is no local response. Removals are communicated to the SLB.

Any new calls that arrive for endpoints that currently have registration entries are not rejected. This is new to this feature with S-CZ7.3.0. Previously, any new call would have been rejected with the **503 Service Unavailable** message. The same **retry-after-upon-offline** action is performed.

Any other SIP methods (like SUBSCRIBE or MESSAGE) intended for this endpoint is handled normally and are not rejected. Priority calls are processed as usual by the SBC, regardless of whether an active registration is present in the SBC as long as the SBC is in **becoming offline** state. When the SBC transitions to the **offline** state, even priority calls are rejected. If the priority calls cannot be forwarded to the endpoint, a **380 Alternative Service** response may be sent, depending on the SBC's configuration. However, when the SBC achieves offline mode, even priority calls are rejected. New non-priority calls coming for endpoints that are not currently registered are rejected with the **503 Service Unavailable** error message, as has always been done.

The SBC sends the endpoint removal requests to the SLB so that the SLB removes them from its endpoint table. If a REGISTER message comes in with multiple contacts, it’s possible that one of the contacts has an active call while others do not. In that scenario, the contact without active call has the Expires value in the Contact header changed to 0 and is forwarded to the core. When the response arrives from the core, the

Contact with active call has its Expires parameter modified to the **retry-after-upon-offline** value or the UA expires value, whichever is lower. Any contact with no active calls is removed from the cache.

Eventually, all SIP calls end, and all registrations expire. The SBC transitions to the **offline** system state. The SBC continues to send heartbeat updates to the SLB.

At any time after the issuance of the **set-system-state offline** command, a **set-system-state online** command may be issued. If the SBC is in **becoming offline** mode, the process is aborted and the SBC again becomes **online**. The SBC state is forwarded to the SLB, and the SBC once again participates in the SLB's (re)balancing process.

2

New for S-Cz7.2.10

This section discusses new Subscriber-Aware Load Balancer (SLB) system capacities included in this release, as well as new and modified commands.

Capacities:

Parameter	Capacity	Configuration Path
Supported endpoints	10,000,000	**setup entitlements**
Supported cluster members	1024	N/A
Supported tunnels	1024	system > network-interface > tunnel-config
Supported policies	16,384	session-router → lb-policy

Acme Packet 4500 SLB ACLI commands that are not supported on the Acme Packet 6100 SLB:

4500 SLB Command	Equivalent 6100 SLB Command	Comments
show service by-index	**show nat in-tabular**	Displays the currently configured datapath service flows.
show service info, show service statistics	**show nat info**	Displays the maximum and current service flow capacity within the platform's datapath layer.
show endpoint by-index	No equivalent 6100 SLB command	Not supported in S-Cx7.2.10.
show endpoint info	No equivalent 6100 SLB command	Not supported in S-Cx7.2.10.

New ACLI Commands

- **setup entitlements**—Configures the number of licensed endpoints on the SLB in increments of 20,000 endpoints. Number must coincide with your Oracle Licensing Agreement.

Caveats for Release S-Cz7.2.10

Note: For Caveats for Session Boarder Controllers (SBCs) when they are Subscriber-Aware Load Balancer (SLB) cluster members, refer to the appropriate documentation for the SBC.

These limitations apply to release S-Cz7.2.10 of the Subscriber-Aware Load Balancer:

Cluster Membership

- Each SBC may be a member of only one cluster, and a cluster may be associated with only one Subscriber-Aware Load Balancer.

Setup Product

- This release officially supports only the SLB product type, and only on the Acme Packet 6100 platform. You will not be able to configure the platform to run as the SBC product type using this release.

Supported Cluster Hardware - This release of software will support cluster members using the following hardware:

- Acme Packet 6100
- Acme Packet 6300
- Acme Packet 4500
- Acme Packet 4600

Supported Cluster Software

- The software release running on the cluster member using IPv4 on the outer tunnel must be one of the following:
  - OCSBC – S-Cx6.2.3
  - OCSBC – S-Cx6.3.0
  - OCSBC – S-Cz7.1.2
  - OCSBC – S-Cz7.2.0
  - OCSBC – S-Cz7.3.0
  - OCUSM – S-Cx6.3.15
  - OCUSM – S-Cz7.1.5
- The software release running on the cluster member using IPv4 on the outer tunnel and IPv6 on the inner tunnel must be one of the following:
  - OCSBC – S-Cz7.1.2
  - OCUSM – S-Cz7.1.5

Protocol Support

- The network-interface tunnel-config protocols TCP and TLS are not supported.
- IMS-AKA is not supported.
- The Oracle Communications Session Border Controller's FTP Server is deprecated. Only SFTP server services are supported.
  - FTP Client access for features such as HDR/CDR push remains.

Fragmented Ping Support

- The Oracle Communications Session Border Controller does not respond to inbound fragmented ping packets.

Physical Interface RTC Support

- After changing any Physical Interface configuration, a system reboot is required.

Packet Trace

- Output from the packet trace local feature on hardware platforms running this software version may display invalid MAC addresses for signaling packets.

High Availability Pairing (HA)

- An Acme Packet 6100 running release S-Cz7.2.10 as a Subscriber-Aware Load Balancer may not be a member of a High Availability pair which includes an Acme Packet 4500.

Known Issues for Release S-Cz7.2.10

Note: For Known Issues for Session Border Controllers (SBC) when they are Subscriber-Aware Load Balancer (SLB) cluster members, refer to the appropriate documentation for the SBC.

These Known Issues apply to release S-Cz7.2.10p3 of the SLB:

Protocols

- Known Issue - L2 lookup miss errors seen. Packets are dropped and cause SIP retransmissions. Seen with TCP endpoint traffic.

Note: This issue is not reproducible as of S-Cz7.2.10p3.

- Known Issue - IPT core miss errors seen. Packets are dropped and there are SIP retransmissions. Normally seen with TCP traffic.

Note: This issue was fixed in S-Cz7.2.10p3.

CPU Utilization

- Known Issue - A memory utilization alarm fires when over 5,000,000 endpoints are balanced.

Note: This issue was fixed in S-Cz7.2.10p3.

Known Issue - An SLB CPU utilization alarm fires and registrations are not rebalanced when load exceeds 2,000 registrations per second.

Balance/Rebalance

- Known Issue - When you add or remove a cluster member in a cluster managed by an HA pair, after the cluster is rebalanced, endpoint entries are not replicated on the standby SLB.

Note: This issue was fixed in S-Cz7.2.10p3.

- Known Issue - The inactivity ageout functionality for trusted endpoints does not work.
- Known Issue - Only a small percentage of endpoint registrations correctly expire and must re-register.

Upgrade from L-Cx1.5.0 configuration files

- Known Issue - When you upgrade from a configuration file created in release L-Cx1.5.0, the parameter **cluster-config>inactive-sd-limit** has an incorrect value after boot.
  - Workaround - Whatever the value of the parameter in the L-Cx1.5.0 config file, after the first boot, the value in the S-Cz7.2.10 file is 1000 times the original value. For example if the value was 1800 (the default value in L-Cx1.5.0), the S-Cz7.2.10 value is 1800000. You can change the value of the **cluster-config>inactive-sd-limit** back to the value in the L-Cx1.5.0 version of the config file and reboot the device, or change the parameter dynamically.

Note: This is a new issue, first noted in S-Cz7.2.10p3.

Dynamic Reconfig

- Known Issue - Dynamic re-configuration of **cluster-config>service-port>protocol** does not remove the service port.

Note: This issue is not reproducible as of S-Cz7.2.10p3.

- Known Issue - Dynamic re-configuration on an SBC does not remove tunnels on the SLB.

Note: This issue is not reproducible as of S-Cz7.2.10p3.

High Availability (HA)

- Known Issue - During a HA failover, IPT core miss errors are incremented, which results in retransmissions.

Note: This is a new issue, first noted in S-Cz7.2.10p3.

- Known Issue - The standby SLB crashes during reboot.

Note: This issue is not reproducible as of S-Cz7.2.10p3.

- Known Issue - The standby SLB does not sync config properly and remains out of service.

Note: This issue was fixed in S-Cz7.2.10p3.

- Known Issue - Standby trusted/untrusted endpoint statistics are incorrect when endpoints are removed.

Note: This issue is not reproducible as of S-Cz7.2.10p3.

- Known Issue - After a switchover, the newly active SLB drops IPv6 endpoints.

Note: This issue is was declared not to be an issue in S-Cz7.2.10p3.

- Known Issue - After a reboot, the standby SLB incorrectly categorizes some synchronized trusted endpoints as untrusted.

Note: This is a new issue, first noted in S-Cz7.2.10p3.

ACLI commands

- Known Issue - The "hits" field of the **show balancer policies** command does not clear after **reset all** is performed.

Note: This issue was fixed in S-Cz7.2.10p3.

- Known Issue - Default values for **source-addr** and **destination-addr** under **lb-policy** are blank.

Note: This issue was fixed in S-Cz7.2.10p2.

Historical data recording (HDR)

Known Issue - A platform running S-Cz7.2.10 as a Subscriber-Aware Load Balancer (SLB) can still configure Session Border Controllers (SBC)-specific group-names.

Note: This is a new issue, first noted in S-Cz7.2.10p3.

Other Known Issues

- Known Issue - Media and management (wancom) interfaces may not be configured with the same subnet, regardless of VLAN.

3

Subscriber-Aware Load Balancer Configuration

SLB Configuration

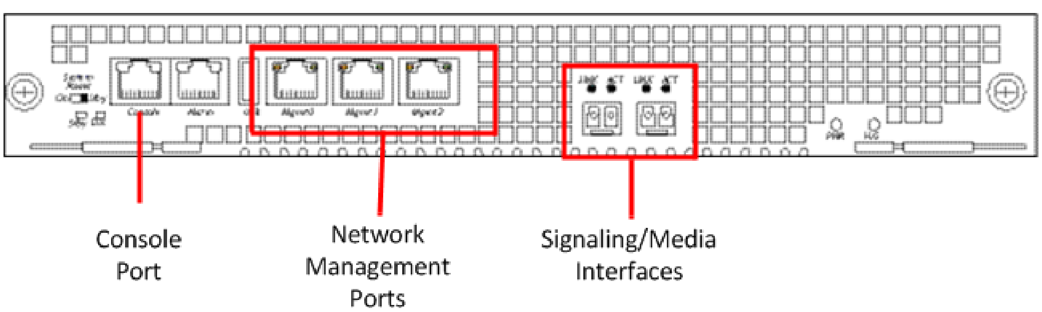
This section explains how to configure functionality specific to the SLB; it does not include configuration steps for functions that it shares in common with its corresponding SBCs (for example, system-config, phy-interface, network-interface, and so on). For information about general SLB configuration, refer to the appropriate documentation as listed in About This Guide.

SLB configuration is quite simple; aside from basic network connectivity, the service interfaces, and the distribution policy, much of the configuration is learned dynamically from the SBCs that comprise the cluster.

Acme Packet 6100 Physical Interfaces

The Acme Packet 6100 supports a single network interface unit (NIU) that contains all external interfaces, including console, alarm, network management and media interfaces. There is currently one type of NIU available, which defines the supported cabling and speed.

The graphic below shows the NIU front panel, which includes all ports and their labeling. This labeling is an important point of reference when you set up the **phy-interface** configuration element.



The Acme Packet 6100 NIU includes the following ports (from left to right).

- Console—Provides serial access for administrative and maintenance purposes.
- Alarm—Dry contact alarm port.
- USB—For use only by Oracle personnel.
- Mgmt0 to Mgmt2—The system uses these 10/100/1000 Base-T Ethernet ports for device management functions. The first interface, Mgmt 0, is for telnet or ssh access to the ACLI. The other two interfaces are

used for state replication for High Availability (HA). For HA, connect these interfaces directly using a crossover cable.

- SFP+ ports—The system uses these 2 x 10GbE ports for signaling and media traffic.

The table below lists the labeling of each interface on the NIU, as well as the applicable **operation-type** and **port** parameters in the **phy-interface** configuration element. Note that the media interfaces are not uniquely labeled with the chassis silkscreen. The table distinguishes between these using "left" and "right", with the perspective being the user looking at the NIU panel.

NIU Label	operation-type	slot	port
Mgmt 0	maintenance	0	0
Mgmt 1	maintenance	0	1
Mgmt 2	maintenance	0	2
USB	NA	NA	NA
NA (left)	media	0	0
NA (right)	media	0	1

Provisioning Entitlements

Provisioning entitlements for the Subscriber-Aware Load Balancer (SLB) is performed by using the **setup entitlements** command.

The **setup entitlements** command is used to configure the total number of endpoints that the SLB has been licensed to manage. The input must be entered in increments of 20,000 endpoints.

A new Acme Packet 6100 platform with no entitlements will boot up for the first time as an SLB product type with 0 endpoints configured. The sample configuration session below shows the steps to finish the configuration of entitlements for the SLB:

```
ACMEPACKET#setup entitlements

----------------------------------------------------------
Entitlements for Subscriber-Aware Load Balancer
Last Modified: Never
----------------------------------------------------------
 1 : LB Endpoint Capacity             : 0

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

  LB Endpoint Capacity (0-10000000)   : 5000000

Enter 1 to modify, d' to display, 's' to save, 'q' to exit. [s]: s
SAVE SUCCEEDED
ural# show entitlements
Provisioned Entitlements:
-------------------------
Subscriber-Aware Load Balancer Base    : enabled
LB Endpoint Capacity                   : 5000000

Keyed (Licensed) Entitlements
-----------------------------
```

After initial configuration using **setup entitlements**, the configuration can be confirmed at any time using **show entitlements**.

SLB Tunnel Configuration

The SLB sends and receives signaling messages to and from clustered SBCs through an IP-in-IP tunnel. The SLB requires one tunnel per interface.

Use the following procedure to perform required SLB-side tunnel configuration. Completion of tunnel configuration is accomplished on the clustered SBCs as described in SBC Tunnel Configuration.

1. From superuser mode, use the following ACLI command sequence to access tunnel-config configuration mode. While in this mode, you partially configure the tunnel-config configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# tunnel-config
ACMEPACKET(tunnel-config)# ?
local-ip-address                tunnel local IP address
port    tunnel                               local & remote control ports
protocol                                              tunnel control
transport protocol
tls-profile                                  tunnel control TLS profile
traffic-policy                      name of traffic-policy for this tunnel
select                                                  select tunnel
to edit
no
delete tunnel
show                                                          show
tunnel
done                                                          write
tunnel information
exit                                                          return
to previous menu
ACMEPACKET(tunnel-config)#
```

2. Use the **local-ip-address** parameter to specify the IP address at the SLB end of the tunnel.

   As the terminus for all tunnels from the clustered SLBs — and never the tunnel originator — only the local address is configured on the SLB.

   Note: This address also supports the exchange of CCP messages.

```
ACMEPACKET(tunnel-config)# local-ip-address 182.16.204.210
ACMEPACKET(tunnel-config)#
```

3. Use the **port** parameter to specify the port used to send and receive CCP messages.

```
ACMEPACKET(tunnel-config)# port 4444
ACMEPACKET(tunnel-config)#
```

4. Use the **protocol** parameter to specify the transport protocol used in support of cluster control messages.

   The only protocol supported for this release is UDP.

```
ACMEPACKET(tunnel-config)# protocol UDP
ACMEPACKET(tunnel-config)#
```

5. Use the **traffic-policy** parameter to specify the traffic-policy for this tunnel.

```
ACMEPACKET(tunnel-config)# traffic-policy tp1
ACMEPACKET(tunnel-config)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.
7. Repeat Steps 1 through 6 to configure additional tunnel-config configuration elements.

Sample SLB Tunnel Configuration

The following formatted extract from **show running-config** ACLI output shows a sample tunnel configuration.

```
tunnel-config
local-ip-address     182.16.204.210
port                 4444
protocol             UDP
tls-profile
last-modified-by     admin@console
last-modified-date   2013-11-07 18:49:04
```

Cluster Configuration

The cluster-config configuration element manages basic SLB interaction with clustered SBCs — it contains a set of global parameters that define the management of the RFC 2003 IP-in-IP tunnels that connect the SLB to clustered SBCs, and the details of rebalance operations. In addition, cluster-config provides for the creation of a list of service interfaces (signaling addresses) that are advertised to endpoints comprising the user access population.

Use the following procedure to perform required cluster-config configuration.

1. From superuser mode, use the following ACLI command sequence to access cluster-config configuration mode. While in this mode, you configure the cluster-config configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# cluster-config
ACMEPACKET(cluster-config)# ?

state                          cluster control state
log-level                      configure log level
auto-rebalance                 Auto-rebalance cluster on new SD availability
source-rebalance-threshold     Percentage of advertised registration capacity
dest-rebalance-threshold       Percentage of advertised registration capacity
dest-rebalance-max             Percentage of advertised registration capacity
tunnel-check-interval          How often an SD's tunnels are checked
tunnel-fail-interval           Time for which no messages have been received
rebalance-request-delay        Delay between subsequent rebalance requests
session-multiplier             ratio of users (endpoints to sessions)
atom-limit-divisor             ratio of atoms (e.g. contacts to endpoints)
rebalance-skip-ahead           Skip endpoints refreshing sooner than
rebalance-max-refresh          Skip endpoints refreshing later than
ignore-tgt-svcs-on-rebalance   When selecting source SDs during rebalancing
rebalance-del-app-entries      Delete Application endpoint Data
inactive-sd-limit              Duration no SD control messages received
                               (seconds)
red-port                       redundant mgcp sync port
red-max-trans                  max redundant transactions to keep
red-sync-start-time            redundant sync start timeout
red-sync-comp-time             redundant sync complete timeout
service-ports                  configure service ports
select                         select cluster config
no                             delete cluster config
show                           show cluster config
done                           save cluster config information
exit                           return to previous menu
```

2. Use the **state** parameter to enable or disable the SLB software.

   The default setting, enabled, enables SLB functionality; disabled renders the SLB inoperable.

```
ACMEPACKET(cluster-config)# state enabled
ACMEPACKET(cluster-config)#
```

3. Use the **log-level** parameter to specify the contents of the SLB log.

   Log messages are listed below in descending order of severity.

   - emergency — the most severe
   - critical
   - major (error)
   - minor (error)
   - warning
   - notice
   - info — (default) the least severe
   - trace — (test/debug, not used in production environments)
   - debug — (test/debug, not used in production environments)
   - detail — (test/debug, not used in production environments)

   In the absence of an explicitly configured value, **log-level** defaults to critical, meaning that log messages with a severity of critical or greater (emergency) are written to the SLB log.

```
ACMEPACKET(cluster-config)# log-level critical
ACMEPACKET(cluster-config)#
```

4. Use the **auto-rebalance** parameter to specify SLB behavior when a new SBC joins an existing cluster.

   With this parameter enabled, the default setting, the SLB redistributes endpoints among cluster members when a new member joins the cluster. Refer to the Rebalancing section for operational details.

   With this parameter disabled, the alternate setting, pre-existing SBCs retain their endpoint populations, and the SLB directs all new endpoints to the newly active SBC until that SBC reaches maximum occupancy.

```
ACMEPACKET(cluster-config)# auto-rebalance enabled
ACMEPACKET(cluster-config)#
```

5. If **auto-rebalance** is set to enabled, use the **source-rebalance-threshold** and **dest-rebalance-threshold** parameters to specify threshold settings that identify existing cluster SBCs as either endpoint sources or endpoint destinations during the rebalance operation. Use the **dest-rebalance-max** parameter to specify the occupancy for the new cluster member. Refer to the Balancing section for details on occupancy and its calculation.

   If **auto-rebalance** is set to disabled, these three parameters can be ignored.

   Parameter values are numeric percentages within the range 0 through 100.

   source-rebalance-threshold specifies the minimum occupancy percent that identifies a clustered SBC as a source of endpoints during a rebalance operation. For example, using the default value of 50 (percent), any clustered SBC with an occupancy rate of 50% or more sheds endpoints during a rebalance. The SLB assigns these endpoints to the new cluster member.

   dest-rebalance-threshold specifies the maximum occupancy percent that identifies a clustered SBC as a destination for endpoints during a rebalance operation. Note that the default setting of 0 (percent), ensures that no pre-existing SBC gains endpoints during a rebalance.

   dest-rebalance-max specifies the maximum occupancy percent that the SLB transfers to the new cluster member during a rebalance operation. The default setting is 80 (percent). Should this threshold value be attained, the SLB distributes remaining endpoints to those SBCs identified as endpoint destinations by their **dest-rebalance-threshold** settings.

```
ACMEPACKET(cluster-config)# source-rebalance-threshold 50
ACMEPACKET(cluster-config)# dest-rebalance-threshold 40
ACMEPACKET(cluster-config)# dest-rebalance-max 75
```

6. If **auto-rebalance** is set to enabled, you can optionally use four additional parameters to fine-tune rebalance operational details.

   If **auto-rebalance** is set to disabled, these four parameters can be ignored.

rebalance-request-delay specifies the interval (in milliseconds) between endpoint request messages sent from the SLB to a clustered SBC. As explained in the Rebalancing section, these messages request a list of endpoints that will be redistributed from the SBC to a new cluster member.

By default, this parameter is set to 500 milliseconds.

Setting this parameter to a higher value results in longer times for the completion of rebalancing; however longer durations provide more time for cluster member processsing of SIP traffic.

rebalance-skip-ahead restricts the target set of SBC endpoints registration eligible for rebalancing to those whose re-registration is not imminent — that is, the registration is not scheduled within the number of milliseconds specified by the parameter setting. Setting this parameter to a non-zero value mitigates against the possibility of a race condition precipitated by a simultaneous endpoint removal generated by the SBC and the arrival of endpoint signalling on an SLB service port. The default setting (0 milliseconds) effectively makes the entire SBC endpoint set eligible for rebalancing.

rebalance-max-refresh restricts the target set of SBC endpoints eligible for rebalancing to those whose re-registration is no further in the future than the time period (milliseconds) specified by this parameter — for example, assuming a parameter value of 6000, the target endpoint set is restricted to those whose re-registration is scheduled within the next 6 seconds.

Because a re-balancing operation necessarily introduces a small window of unreachability for re-balanced endpoints, this parameter provides users with some degree of control over the period of time that a re-balanced endpoint may be unreachable.

The default setting (0 milliseconds) effectively makes the entire SBC endpoint set eligible for rebalancing.

rebalance-del-app-entries specifies when cached SIP entries for rebalanced endpoints are removed from the clustered SBC. The default setting (disabled) specifies that cached entries are retained after a rebalance operation, and subsequently removed from the cache by standard time-out procedures. When set to enabled, this parameter specifies that the SBC removes cached registration entries at the completion of the rebalance operation.

```
ACMEPACKET(cluster-config)# rebalance-request-delay 750
ACMEPACKET(cluster-config)# rebalance-skip-ahead 100
ACMEPACKET(cluster-config)# rebalance-max-refresh 1000
ACMEPACKET(cluster-config)# rebalance-del-app-entries enabled
```

7. Three parameters, **tunnel-fail-interval**, **tunnel-check-interval,** and **inactive-sd-limit** maintain and monitor the IP-in-IP tunnels established between the SLB and clustered SBCSLBs.

   tunnel-fail-interval specifies the interval (in milliseconds) between periodic keepalive messages sent from a clustered SBC to the SLB. If the SLB fails to receive a keepalive message within the specified period, it flags the tunnel as dead. By default, this parameter is set to 10000 milliseconds.

   tunnel-check-interval specifies the interval (in milliseconds) between SLB tunnel audits. During a tunnel audit, the SLB checks the status of each tunnel and removes all tunnels flagged as dead. If all of a cluster member's tunnels are removed. the SLB places that cluster member in an out-of-service state. By default, this parameter is set to 15000 milliseconds.

   If you change default settings for either parameter, ensure that the setting for **tunnel-check-interval** is greater than the **tunnel-fail-interval** setting.

   inactive-sd-limit specifies the maximum silent interval (defined as the absence of heartbeat traffic from any tunnel) seconds) before the SLB flags a cluster member as dead, and removes that SBC from the cluster. By default, this parameter is set to 1800 seconds (30 minutes). supported values are integers within the range 0 through 31556926 (365 days).

```
ACMEPACKET(cluster-config)# tunnel-fail-interval 10000
ACMEPACKET(cluster-config)# tunnel-check-interval 15000
ACMEPACKET(cluster-config)# inactive-sd-limit 900
```

8. Use the **session-multiplier** and **atom-limit-divisor** parameters to specify optional, user-configurable numeric factors used in occupancy and occupancy rate calculations.

session-multiplier provides a factor that when multiplied by an SBC's licensed session limit, determines the maximum number of endpoints that the SBC can support (that is, its maximum occupancy).

The default setting is 10; valid settings include any integer values within the range 1 through 100.

Using the default setting, an SBC licensed for 32,000 concurrent sessions has a maximum theoretical occupancy of 320,000 endpoints.

atom-limit-divisor provides another factor that can be used in occupancy and occupancy percent calculations. By default, occupancy calculations are based on endpoints (IP addresses), and do not take into account the fact that the same IP address can represent multiple users.

The default setting is 1, which assumes a conservative 1-to-1 correlation between endpoints and users; valid settings include any integer values within the range 1 through 1000.

Note: The SLB initially calculates a tentative maximum occupancy value, expressed as a number of endpoint addresses, for each clustered SBC. SLB calculations are based upon the licensed capacity of each cluster member, and the values assigned to the session-multiplier and atom-limit-divisor parameters. After calculating the tentative maximum occupancy value, the SLB compares this value to the value of the registration-cache-limit parameter as defined on the clustered SBC. If the value of registration-cache-limit is either 0, or greater than the tentative maximum occupancy value, the calculated value is retained as the occupancy ceiling. However, if the registration-cache-limit value is greater than 0, but less than the tentative calculation, the value of registration-cache-limit is used as the occupancy ceiling.

Once an SBC has reached its maximum number of endpoints, the SLB removes it from the load balancing algorithm. These parameter settings should be changed only after careful examination of network conditions and behavior.

```
ACMEPACKET(cluster-config)# session-multiplier 10
ACMEPACKET(cluster-config)# atom-limit-divisor 1
```

9. The **ignore-tgt-svc-on-rebalance** parameter is not currently supported, and can be safely ignored.
10. Retain default settings for the **red-port**, **red-max-trans**, **red-sync-start-time**, and **red-sync-comp-time** parameters.
11. Use **done**, **exit**, and **verify-config** to complete cluster configuration.

Sample Cluster Configuration

The following formatted extract from **show running-config** ACLI output shows a sample cluster configuration.

```
cluster-config
state                           enabled
log-level                       CRITICAL
auto-rebalance                  enabled
source-rebalance-threshold      50
dest-rebalance-threshold        40
dest-rebalance-max              75
tunnel-check-interval           750
tunnel-fail-interval            10000
rebalance-request-delay         500
session-multiplier              4
rebalance-skip-ahead            0
rebalance-max-refresh           0
ignore-tgt-svcs-on-rebalance    disabled
atom-limit-divisor              1000
rebalance-del-app-entries       disabled
inactive-sd-limit               1800
red-port                        2001
red-max-trans                   10000
red-sync-start-time             5000
red-sync-comp-time              1000
```

```
service-port
last-modified-by                admin@console
last-modified-date              2013-11-07 18:49:04
```

Service Ports Configuration

A service port is essentially a SIP port monitored by the SLB for incoming signaling from the user population. For virtually all network topologies, multiple service ports are expected on a typical SLB configuration. A service-port is a multiple instance configuration element; for each service port advertised to the access network(s), at least one service-port configuration element must be configured.

Use the following procedure to perform required service-ports configuration.

1. From superuser mode, use the following ACLI command sequence to access service-port configuration mode. While in this mode, you configure one or more service-port configuration elements.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# cluster-config
ACMEPACKET(cluster-config)# service-ports
ACMEPACKET(service-port)# ?
address                    IP address
port                       port (default: 5060)
protocol                   transport protocol
network-interface          network interface for service port
select                     select cluster config
no                         delete cluster config
show                       show cluster config
done                       save cluster config information
exit                       return to previous menu
ACMEPACKET(service-port)#
```

2. Use the required **address** parameter to specify the IPv4 or IPv6 address of this service port.

```
ACMEPACKET(service-port)# address 10.0.0.1
ACMEPACKET(service-port)#
```

3. Use the **port** parameter to specify the port monitored by the SLB for incoming signaling messages.

   In the absence of an explicitly configured port, the SLB provides a default value of 5060 (the registered SIP port).

   Allowable values are integers within the range 0 through 65535.

```
ACMEPACKET(service-port)# port 5060
ACMEPACKET(service-port)#
```

4. Use the **protocol** parameter to choose the transport protocol.

   The supported setting is UDP (the recommended default).

```
ACMEPACKET(service-port)# protocol udp
ACMEPACKET(service-port)#
```

5. Use the required **network-interface** parameter to identify the SLB network interface that supports this service port. With this parameter, you have the option of specifying IPv4 or IPv6 (.4 or .6).

```
ACMEPACKET(service-port)# network-interface M00:0.4
ACMEPACKET(service-port)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of this service-port configuration element.
7. Repeat Steps 1 through 6 to configure additional service-port configuration elements.

Sample Service Port Configuration

The following formatted extract from **show running-config** ACLI output shows a sample service port configuration.

```
service-port
address                         192.169.203.83
port                            5060
protocol                        UDP
network-interface               M00:0.4
last-modified-by                admin@console
last-modified-date              2013-11-07 18:49:04
```

Traffic Policy Configuration

This configuration record will enable management of tunnel bandwidth on a per-cluster member basis. The **name** of this policy will be entered into the SBC Tunnel Configurations.

Note: If you do not need to change any of the implicit defaults for the traffic policy, you do not need to configure this policy at all. The implicit default configuration for this policy is as below. If you must change any of the parameters from the implicit default, you must name the resulting traffic policy **`default`**.

Use the following procedure to perform traffic policy configuration if required.

1. Access the **traffic-policy-config** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# traffic-policy-config
ACMEPACKET(traffic-policy-config)#
name        throttle-rate     max-signaling-rate     min-untrusted-pct
max-untrusted-pct      options      select      no
show       done       quit       exit
```

2. Enter a **name** for this traffic policy configuration. This is the string identifier for this policy.

```
ACMEPACKET(traffic-policy-config)# name  tp1
ACMEPACKET(traffic-policy-config)#
```

3. Enter a **throttle-rate** for this traffic policy configuration. This is the host throttle rate in registrations per second.

```
ACMEPACKET(traffic-policy-config)# throttle-rate 800
ACMEPACKET(traffic-policy-config)#
```

4. Enter a **max-signaling-rate** for this traffic policy configuration. This is the maximum signaling rate to a cluster member in bytes per second.

```
ACMEPACKET(traffic-policy-config)# max-signaling-rate 33000000
ACMEPACKET(traffic-policy-config)#
```

5. Enter a **min-untrusted-pct** for this traffic policy configuration. This is the minimum percentage of signaling rate allocated to untrusted traffic.

```
ACMEPACKET(traffic-policy-config)# min-untrusted-pct 33
ACMEPACKET(traffic-policy-config)#
```

6. Enter a **max-untrusted-pct** for this traffic policy configuration. This is the maximum percentage of signaling rate allocated to untrusted traffic.

```
ACMEPACKET(traffic-policy-config)# max-untrusted-pct 66
ACMEPACKET(traffic-policy-config)#
```

7. Use **done**, **exit**, and **verify-config** to complete configuration of this traffic policy configuration element.
8. Repeat steps 1 through 7 to configure additional traffic policy configuration elements.

Sample Traffic Policy Configuration

The following formatted extract from traffic-policy-config shows the default policy configuration.

Note: For this initial release of software, if you do not need to change any of the implicit defaults for the traffic policy, you do not need to configure this policy at all. The implicit default configuration for this policy is as below. If you need to change any of the parameters from the implicit default, you must name the resulting traffic policy **`default`**

```
ACMEPACKET# traffic-policy-config
name
      default
throttle-rate                                         800
max-signaling-rate                              33000000000
min-untrusted-percent          33
max-untrusted-percent          66
```

Load Balancer Policy Configuration

The lbp-config configuration element manages the SLB endpoint table. It also creates and manages a list of service interfaces (signaling addresses) that are advertised to endpoints comprising the user access population.

Use the following procedure to perform required lbp-config configuration.

1. From superuser mode, use the following ACLI command sequence to access lbp-config configuration mode. While in this mode, you configure the lbp-config configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# lbp-config
ACMEPACKET(lbp-config)#?
state                                lbp state
log-level                            configure log level
untrusted-grace-period               Untrusted grace period
max-untrusted-percentage             Maximum untrusted endpoints percentage
max-untrusted-upper-threshold        Maximum untrusted endpoints upper
                                     threshold
max-untrusted-lower-threshold        Maximum untrusted endpoints upper
                                     threshold
endpoint-capacity-upper-threshold    endpoint capacity upper threshold
endpoint-capacity-lower-threshold    endpoint capacity lower threshold
red-port                             lbp redundant sync port: 0 to disable
                                     and 2000 to enable
red-max-trans                        maximum redundancy transactions to keep
                                     on active
red-sync-start-time                  timeout for transitioning from standby
                                     to active
red-sync-comp-time                   sync request timeout after initial sync
                                     completion
port-aware-balancing                 Include endpoint source port, in
addition to the source IP address if NAT is used
options                              optional features/parameters
select                               select lbp config
no                                   delete lbp config
show                                 show lbp config
done                                 save lbp config information
exit                                 return to previous menu
ACMEPACKET(lbp-config)#
```

2. Use the **state** parameter to enable or disable the SLB software.

   The default setting, enabled, enables SLB functionality; disabled renders the SLB inoperable.

```
ACMEPACKET(lbp-config)# state enable
ACMEPACKET(lbp-config)#
```

3. Use the **log-level** parameter to specify the contents of the SLB log.

   Log messages are listed below in descending order of severity.

   - emergency — the most severe
   - critical
   - major (error)
   - minor (error)

- warning
- notice
- info — (default) the least severe
- trace — (test/debug, not used in production environments)
- debug — (test/debug, not used in production environments)
- detail — (test/debug, not used in production environments)

In the absence of an explicitly configured value, **log-level** defaults to critical, meaning that log messages with a severity of critical or greater (emergency) are written to the LBP log.

```
ACMEPACKET(lbp-config)# log-level critical
ACMEPACKET(lbp-config)#
```

4. Use the **untrusted-grace-period**, **max-untrusted-percentage**, **max-untrusted-upper-threshold**, and **max-untrusted-lower-threshold** parameters to implement percentage-based management and monitoring of untrusted endpoints in the SLB endpoint database. Management and monitoring of untrusted endpoints is instrumental in detecting and responding to Denial-of-Service (DOS) attacks aimed at the SLB.

   untrusted-grace-period specifies the maximum time, in seconds, that a forwarding rule is retained by the SLB before it is confirmed with a promotion message from the SBC that received the untrusted endpoint. Refer to the Balancing section for message details.

   In the absence of an explicitly assigned value, the SLB provides a default setting of 30 (seconds).

   If this time period elapses without a promotion message arriving to confirm this user, the SLB deletes the entry.

   Setting this parameter to 0 allows untrusted/unconfirmed entries to exist indefinitely without aging out.

   max-untrusted-percentage specifies the percentage of the overall endpoint population that is reserved for untrusted users.

   The default setting is 20 (percent); supported values are integers within the range 1 through 100.

   This percentage is applied to the overall remaining occupancy of the SLB after trusted (confirmed) users are accounted for. For example, when empty, the SLB holds two million forwarding rules; assuming the default setting, at most 400,000 rules are reserved for untrusted rules. By the time one million users have been promoted, 20% of the remaining space means that up to 200,000 entries can be used for untrusted users.

   max-untrusted-upper-threshold specifies a threshold level at which the SLB (1) raises an alarm, and (2) issues an SNMP trap reporting an excessive number of untrusted endpoints within the entire endpoint population.

   This parameter, which has a default setting of 80 (percent), is calculated as a percent of **max-untrusted-percentage**. For example, assuming default settings for both parameters, the SLB raises an alarm and issues an SNMP trap when the percentage of untrusted endpoints attains 16%.

   max-untrusted-lower-threshold specifies a threshold level at which the SLB (1) clears the existing untrusted endpoint alarm, and (2) issues an SNMP trap reporting alarm clearance.

   This parameter, which has a default setting of 70 (percent), is calculated as a percent of **max-untrusted-percentage**. For example, assuming default settings for both parameters, the SLB clears an alarm and issues an SNMP trap when the percentage of untrusted endpoints falls to 14%.

```
ACMEPACKET(lbp-config)# untrusted-grace-period 30
ACMEPACKET(lbp-config)# max-untrusted-percentage 20
ACMEPACKET(lbp-config)# max-untrusted-upper-threshold 80
ACMEPACKET(lbp-config)# max-untrusted-lower-threshold 70
ACMEPACKET(lbp-config)#
```

5. Use the **endpoint-capacity-upper-threshold** and **endpoint-capacity-lower-threshold** parameters to implement license-based management and monitoring of the SLB endpoint counts.

endpoint-capacity-upper-threshold specifies a threshold level at which the SLB (1) raises an alarm, and (2) issues an SNMP trap reporting an excessive number of active endpoints.

This parameter, which has a default setting of 80 (percent), is calculated as a percentage of the endpoints allowed by the installed SLB license.

endpoint-capacity-lower-threshold specifies a threshold level at which the SLB (1) clears the existing endpoint alarm, and (2) issues an SNMP trap reporting alarm clearance.

This parameter, which has a default setting of 70 (percent), is calculated as a percentage of the endpoints allowed by the installed SLB license.

```
ACMEPACKET(lbp-config)# endpoint-capacity-upper-threshold 80
ACMEPACKET(lbp-config)# endpoint-capacity-lower-threshold 70
ACMEPACKET(lbp-config)#
```

6. Enable **port-aware-balancing** to include endpoint source port, in addition to the source IP and destination service representation when looking up a unique EPT prior to forwarding towards the SBC cluster. Choices are enabled and disabled. Default is disabled.

```
ACMEPACKET(lbp-config)# port-aware-balancing enable
ACMEPACKET(lbp-config)#
```

7. Use **done**, **exit**, and **verify-config** to complete configuration of this load-balancer-policy configuration element.

Sample Load Balancer Policy Configuration

The following formatted extract from **show running-config** ACLI output shows a sample load balancer policy configuration with port-aware-balancing enabled.

```
lbp-config
state                                enabled
log-level                            NOTICE
untrusted-grace-period               30
max-untrusted-percentage             20
max-untrusted-upper-threshold        80
max-untrusted-lower-threshold        70
end-point-capacity-upper-threshold   80
end-point-capacity-lower-threshold   70
red-port                             0
red-max-trans                        500000
red-sync-start-time                  5000
red-sync-comp-time                   1000
port-aware-balancing                 enabled
last-modified-by                     admin@console
last-modified-date                   2015-11-07 18:49:04
```

Distribution Policy Configuration

Distributing endpoints equitably among the cluster members is the primary function of the SLB. The lp-config configuration element allows you to control the method of the SLB’s distribution based on matching criteria. Using inbound packet matching criteria, you can control the assignment of users to SBCs. Matching is done by data available up to and including the transport layer of the packet: source IP address and port, destination IP address and port, and transport protocol. The IP addresses and ports may or may not include bit masks as well.

Conceptually, the load balancer policy table, with sample data, looks akin to the following.

Source IP/Mask	Source Port/ Mask	Destination IP/ Mask	Destination Port/Mask	Transport Protocol Requirements (list)	Realm Identifiers (list)
192.168.7.22/32	0/0	10.0.0.1/32	5060/16		West

Source IP/Mask	Source Port/ Mask	Destination IP/ Mask	Destination Port/Mask	Transport Protocol Requirements (list)	Realm Identifiers (list)
192.168.1.0/24	0/0	10.0.0.1/32	5060/16	UDP, TCP	North, South, West
192.168.0.0/16	0/0	10.0.0.1/32	5060/16	UDP, TCP	East, West
0.0.0.0/0	0/0	0.0.0.0/0	0/0		

Policies are matched using a longest prefix match algorithm; the most specific policy is selected when comparing policies to received packets. One and only one policy is chosen per packet; if the next hops in that route are all unavailable, the next best route is not consulted (instead, the default policy may be consulted – see below). This is different than the local-policy behavior on the SBC.

Within each policy you may configure multiple next hops, where each next hop is a named group of SBCs. In the sample policy table, this is indicated in the second policy with a source IP range of 192.168.1.0/24. The realm identifier list for this policy indicates North, South, West. Each of these realm identifiers represents a collection of zero or more SBCs, in SBC parlance these are roughly analogous to session-agent groups. Each of these realm identifiers is also assigned a priority (a value between 1 and 31, with 31 representing the highest priority) in the configuration, and the SLB sorts the possible destinations with the highest priority first. Upon receipt of a packet matching a policy with multiple configured realm identifiers, the SLB gives preference to SBCs from the realm identifier with the highest priority. Should no SBCs be available in that priority level (due to saturation, unavailability, and so on.) the SLB moves on to investigate the next priority level, and so on. Should no SBCs become available after traversing the entire list of all SBCs within each priority level, the SBC either drops the packet or attempt to use the default policy.

The bottom row of the sample table shows this implicit, last resort default policy. When enabled, the SLB reverts to the default policy when all of the potential next hop realms referenced in the endpoint's distribution rule are unavailable. In that event, the default policy attempts to locate a clustered SBC that advertises support for the service-interface that the packet arrived on. The realm is not considered when matching to the default policy. If such an SBC is found, the SLB forwards the packet to that DBC; if such an SBC is not found, the SLB drops the packet.

It is not necessary to configure the default policy — it is simply intended as a catchall policy, and may be used when all that is required is a simple round-robin balancing scheme based on simple metrics (for example, CPU utilization and number of registrations currently hosted by an SBC). If no policies are configured on the SLB, the default policy is used. The default realm is implied in the above table as * and is enabled by default for policy records.

Use the following procedure to perform required lb-config configuration.

1. From superuser mode, use the following ACLI command sequence to access lb-config configuration mode. While in this mode, you configure the distribution rules used to implement policy-based load balancing on the SLB.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# lb-policy
ACMEPACKET(lb-policy)# ?
state                   lb policy state
default-realm           use default realm
description             load balancer policy description
protocols               list of protocols
lb-realms               list of realms
                               name
                               priority
source-addr             source ip address
```

```
destination-addr        destination ip address
select                  select lb policy
no                      delete lb policy
show                    show lb policy
done                    save lb policy information
exit                    return to previous menu
ACMEPACKET(lb-policy)#
```

2. Use the **state** parameter to enable or disable this distribution rule.

   The default setting, enabled, enables the distribution rule; disabled disables the rule.

```
ACMEPACKET(lb-policy)# state enabled
ACMEPACKET(lb-policy)#
```

3. Use the **default-realm** parameter to enable or disable the default distribution policy.

   The default setting, enabled, enables the default policy; disabled disables the policy.

   With **default-realm** enabled, the SLB provides a best-effort delivery model if the next-hop realms listed in this distribution rule are unavailable. With **default-realm** disabled, the orphaned packet is dropped.

```
ACMEPACKET(lb-policy)# default-realm enabled
ACMEPACKET(lb-policy)#
```

4. Optionally use the **description** parameter to provide a description of this distribution rule.

```
ACMEPACKET(lb-policy)# description Local traffic to Los Angeles site
ACMEPACKET(lb-policy)#
```

5. Use the **protocols** parameter to construct a list of protocols that must be supported by this distribution rule.

```
ACMEPACKET(lb-policy)# protocols udp
ACMEPACKET(lb-policy)#
```

6. Use either the **source-addr** parameter or the **destination-address** parameter to specify matching criteria for this distribution rule.

   Use the **source-addr** parameter to specify source-address-based matching criteria.

   Packets whose source IP addresses match the criteria specified by this parameter are subject to this distribution rule.

```
ACMEPACKET(lb-policy)# source-addr 10.0.0.1
ACMEPACKET(lb-policy)#
```

   matches any port on the specified IP source address

```
ACMEPACKET(lb-policy)# source-addr 10.0.0.1:5060
ACMEPACKET(lb-policy)#
```

   matches the specified IP source address:port pair

```
ACMEPACKET(lb-policy)# source-addr 10.0.0.1/24
ACMEPACKET(lb-policy)#
```

   matches any IP source address, any port on the 10.0.0.x subnet

```
ACMEPACKET(lb-policy)# source-addr 10.0.0.240/28:5060
ACMEPACKET(lb-policy)#
```

   matches IP source addresses 10.0.0.240:5060 through 10.0.0.255:5060

   Use the **destination-addr** parameter to specify destination-address-based matching criteria.

   Packets whose destination IP addresses match the criteria specified by this parameter are subject to this distribution rule.

```
ACMEPACKET(lb-policy)# destination-addr 10.0.0.1
ACMEPACKET(lb-policy)#
```

   matches any port on the specified IP destination address

```
ACMEPACKET(lb-policy)# destination-addr 10.0.0.1:5060
ACMEPACKET(lb-policy)#
```

matches the specified IP destination address:port pair

```
ACMEPACKET(lb-policy)# destination-addr 10.0.0.1/24
ACMEPACKET(lb-policy)#
```

matches any IP destination address, any port on the 10.0.0.x subnet

```
ACMEPACKET(lb-policy)# destination-addr 10.0.0.240/28:5060
ACMEPACKET(lb-policy)#
```

matches destination IP addresses 10.0.0.240:5060 through 10.0.0.255:5060

7. Use the **lb-realms** parameter to access lb-realm configuration mode.

   While in lb-realm configuration mode you identify one or more SLBs eligible to receive traffic that matches this distribution rule.

```
ACMEPACKET(lb-policy)# lb-realms
ACMEPACKET(lb-realm)#
name              realm name (string identifier)
priority          priority (range 1-31)
select            select a lb realm to edit
no                delete selected lb realm
show              show lb realm information
done              write lb realm information
exit              return to previous menu
ACMEPACKET(lb-realm)#
```

8. Use the **name** parameter to identify the realm.

   As previously discussed, the name field is roughly analogous to an SBC session-agent group. SBCs configured to communicate within a cluster hosted by an SLB advertise offered services to the SLB. These services (for example, SIP support) exist in realms, whose names are sent to the SLB as part of the SBC advertisement. The SLB, upon receipt of these advertisements, joins each SBC into one or more realm identifier groups based upon the realm name(s) the SBC has offered up. The **name** command of the lb-realm configuration element matches this distribution rule to a supporting SBC that has offered that realm name for cluster membership.

```
ACMEPACKET(lb-realm)# name LosAngeles
ACMEPACKET(lb-realm)#
```

9. Use the **priority** parameter to specify the realm priority.

   Priority is expressed as an integer value within the range 0 to 31 — the higher the integer, the greater the priority.

   The default value, 0, specifies use of the default routing policy, and should not be used when policy-based distribution is enabled.

   Priority values are considered when multiple SBCs offer the same service to matched packets.

```
ACMEPACKET(lb-realm)# priority 31
ACMEPACKET(lb-realm)#
```

10. Use **done**, **exit**, and **verify-config** to complete configuration of this lb-realm configuration element.
11. To specify other eligible SLBs, repeat Steps 7 through 10. For example,

```
ACMEPACKET(lb-config)# lb-realms
ACMEPACKET(lb-realm)# name LasVegas
ACMEPACKET(lb-realm)# priority 25
ACMEPACKET(lb-realm)# done
ACMEPACKET(lb-realm)# exit
ACMEPACKET(lb-realm)# verify-config
```

12. Use **done**, **exit**, and **verify-config** to complete configuration of this distribution rule.
13. To specify additional distribution rules, repeat Steps 1 through 12 as often as necessary.

Sample Distribution Rule Configurations

The following formatted extract from **show running-config** ACLI output shows sample distribution rule configurations.

```
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
       lb-realm
       name          Realm192p1
       priority      10
source-addr          1.1.0.0/16
destination-addr     0.0.0.0/0
last-modified-by     admin@console
last-modified-date   2013-11-07 18:58:10
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
       lb-realm
       name          Realm192p1
       priority      7
source-addr          1.20.0.0/16
destination-addr     0.0.0.0/0
last-modified-by     admin@console
last-modified-date   2013-11-07 19:01:01
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
       lb-realm
       name          Realm192p1
       priority      5
source-addr          1.120.0.0/16
destination-addr     0.0.0.0/0
last-modified-by     admin@console
last-modified-date   2013-11-07 19:00:49
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
       lb-realm
       name          Realm192p1
       priority      3
```

Forced Rebalance

The **notify ccd rebalance** ACLI command initiates an immediate forced rebalance operation. A forced rebalance operation is identical to the one described in the Rebalancing section.

notify ccd rebalance [cancel [sd-name]]

```
ACMEPACKET# notify ccd rebalance
```

initiates the forced rebalance by calculating drop counts for each eligible cluster member, and then requesting drops from the first cluster member in the rebalance queue.

```
ACMEPACKET# notify ccd rebalance cancel
```

terminates the forced rebalance.

```
ACMEPACKET# notify ccd rebalance cancel ~sam
```

terminates the forced rebalance for a specified cluster member. Note the use of tilde special character, which forces the SLB to do a substring match of the following string against all cluster member names. Assuming a cluster member samadams@172.30.68.31 — that cluster member removes itself from the rebalance queue, if it has not yet removed endpoints, or ceases endpoint removal and exits the queue if it is currently doing so.

The **notify ccd drop** ACLI command instructs the target cluster member to drop a specific number of endpoints from a specific realm, from all realms, or without regard for realm.

notify ccd drop <sd-name> (<realm> <number> | <number>)

```
ACMEPACKET# notify ccd drop ~sam boston 100
```

instructs the target cluster member to drop 100 endpoints from the boston realm

```
ACMEPACKET# notify ccd drop ~sam * 100
```

using the * special character instructs the target cluster member to drop 100 endpoints from all realms

```
ACMEPACKET# notify ccd drop ~sam 100
```

instructs the target cluster member to drop 100 endpoints without regard for realm

SBC Configuration

This section describes the configuration necessary to allow an SBC to join a cluster. Configuration is simplified to allow for an easy and seamless migration from a deployed standalone SBC to a deployed clustered SBC. There are only two places where new configuration is required: in the network-interface configuration element, where tunnel information is defined; and in the signaling application’s interface, (the sip-interface configuration element).

SBC Tunnel Configuration

Configuring the properties of the IP-in-IP tunnel on the SBC is a matter of configuring the local IP address, remote IP address, and specifying transport layer and application layer protocol support.

The following example uses a tunnel named sipSignaling, which was initially and partially configured on the SLB. Note in the following configuration that the value of **remote-ip-address** parameter must agree with the value which was previously set with the **local-ip-address** parameter on the SLB. The complementary configuration performed on the SLB enables tunnel establishment between the SBC and the SLB.

1. From superuser mode, use the following ACLI command sequence to access tunnel-config configuration mode. While in this mode, you perform required SLB tunnel configuration.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# tunnel-config
ACMEPACKET(tunnel-config)# ?
name                    tunnel name
local-ip-address        tunnel local IP address
remote-mac-address      tunnel remote mac address
remote-ip-address       tunnel remote IP address
application             application protocol for this tunnel
port                    tunnel local & remote control ports
protocol                tunnel control transport protocol
tls-profile             tunnel control TLS profile
traffic-policy                                Name of traffic policy that

            applies to this tunnel
```

```
select                    select tunnel to edit
no                        delete tunnel
show                      show tunnel
done                      write tunnel information
exit                      return to previous menu
ACMEPACKET(tunnel-config)#
```

2. Use the **name** command to provide a unique identifier for this tunnel instance.

```
ACMEPACKET(tunnel-config)# name sipSignaling
ACMEPACKET(tunnel-config)#
```

3. Use the **local-ip-address** parameter to specify the IP address at the SLB end of the tunnel.

   Note: This address also supports the exchange of CCP messages.

```
ACMEPACKET(tunnel-config)# local-ip-address 1.1.1.100
ACMEPACKET(tunnel-config)#
```

4. Ignore the **remote-mac-address** parameter which is not required for tunnel configuration.
5. Use the **remote-ip-address** parameter to specify the IP address at the SLB end of the tunnel.

   Note: This address also supports the exchange of CCP messages.

```
ACMEPACKET(tunnel-config)# remote-ip-address 182.16.204.210
ACMEPACKET(tunnel-config)#
```

6. Use the **port** parameter to specify the port used to send and receive cluster control messages.

```
ACMEPACKET(tunnel-config)# port 4444
ACMEPACKET(tunnel-config)#
```

7. Use the **protocol** parameter to specify the transport protocol used in support of cluster control messages.

   Supported transport protocol is UDP (the recommended default).

```
ACMEPACKET(tunnel-config)# protocol UDP
ACMEPACKET(tunnel-config)#
```

8. Use the **application** parameter to specify the application protocol supported by this tunnel.

   Specify the SIP protocol.

```
ACMEPACKET(tunnel-config)# application SIP
ACMEPACKET(tunnel-config)#
```

9. Use **traffic-policy** to enter the name of the traffic policy that applies to this tunnel (1-128 characters long) as configured on the SLB

   This configuration is a per-tunnel configuration. Once configured, it will be passed on via the CCP protocol to SLB in Heartbeat messages.

   The CCD task running on the SLB will extract the traffic policy name and will find the matching traffic-policy configuration on the SLB.

```
ACMEPACKET(tunnel-config)# traffic-policy <pattern>
ACMEPACKET(tunnel-config)#
```

10. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.
11. Repeat Steps 1 through 9 to complete tunnel configuration on other SIP interfaces as required.

Sample SBC Tunnel Configuration

The following formatted extract from **show running-config** ACLI output shows a sample SLB (cluster member) configuration.

```
tunnel-config
name                               one
local-ip-address                   1.1.1.100
```

```
remote-mac-address
remote-ip-address
182.16.204.210
port                           4444
protocol                       UDP
tls-
profile
         TLS-LB
traffic-
policy                                                                     tp1
application                    SIP
last-modified-by               admin@console
last-modified-date             2013-11-10 23:24:15
```

Note: This configuration is a per-tunnel configuration. Once configured, it will be passed on via the CCP protocol to SLB in Heartbeat messages.

SIP Configuration

In a traditional SBC configuration the IP address assigned to a sip-port configuration element is contained within the address space defined by the network interface netmask. This is not be the case for clustered SBCs. Rather, the IP address assigned to the sip-port is identical to the address of an SLB service-port advertised on the access network. The process of encapsulating the packets between the SLB and SBC masks the fact that the IP address the SBC expects to receive IP packets on is different than the Layer 5 address the SBC expects the SIP address on.

Consistency of realm identification is vital to successful and predictable policy-based load balancing. Take particular care to ensure that the **realm-id** of the sip-interface configuration element mirrors the **lb-realm** assignments made while configuring distribution rules. See the Distribution Policy Configuration section.

In the following configuration example, the **realm-id** is LosAngeles. This SBC, when booted, will detect that it is a member of an SLB cluster and register the service port 10.0.0.1:5060/UDP as the realm LosAngeles with the SLB. The SLB will automatically create the SBC group LosAngeles (if it doesn't exist) or join the SBC to the group LosAngeles (if it is not the first to advertise LosAngeles). Policy statements that direct packets to LosAngeles now consider this SBC as a potential destination, assuming the address:port/protocol also are consistent with the policy's matching criteria.

This technique allows you to configure the same IP:port/protocol on multiple SBCs, with different realm-id labels, to indicate priority of one SBC or group of SBCs over another. As an example, consider several SBCs geographically situated together with the label LosAngeles, and several other SBCs geographically situated elsewhere with the label NewYork, all with the identical SIP interface and SIP port configuration. A policy can be easily defined to give preference to a source subnet of users in California to the LosAngeles member SBCs, with NewYork as a second priority. This provides flexibility in network design without undue burden in the configuration: SBCs' tagged with the same realm name are joined in dynamically created SBC groups by the SLB, with no explicit configuration required on the SLB whatsoever.

1. From superuser mode, use the following ACLI command sequence to access sip-interface configuration mode. While in this mode, you verify the **realm-id** and assign the newly created IP-in-IP tunnel to a SIP interface.

```
westy# configure terminal
westy(configure)# session-router
westy(session-router)# sip-interface
westy(sip-interface)# select
<realm-id>: LosAngeles
1: LosAngeles  172.192.1.15:5060
selection: 1
westy(sip-interface)# show
sip-interface
        state                    enabled
        realm-id                 LosAngeles
        ...
```

```
        ...
        ...
westy(sip-interface)#
```

2. Use the **tunnel-name** parameter to assign the IP-in-IP tunnel to the current SIP interface.

```
westy(sip-interface)# tunnel-name sipSignaling
westy(sip-interface)# ?
```

3. Use the **sip-port** command to move to sip-port configuration mode.

```
westy(sip-interface)# sip-port
westy(sip-port)# ?
address            IP Address
port               port (default: 5060)
transport-protocol transport protocol
tls-profile        the profile name
allow-anonymous    allowed requests from SIP realm
ims-aka-profile    ims-aka profile name
select             select a sip port to edit
no                 delete a selected sip port
show               show sip port information
done               write sip port information
exit               return to previous menu
westy(sip-port)#
```

4. Use the **address**, **port**, and **transport-protocol** parameters to mirror the address of an existing SLB service port.

```
westy(sip-port)# address 10.0.0.1
westy(sip-port)# port 5060
westy(sip-port)# transport-protocol udp
westy(sip-port)#
```

5. Use **done**, **exit**, and **verify-config** to complete configuration of this sip-port configuration element.
6. Repeat Steps 1 through 5 as necessary to verify **realm-id**s, assign IP-in-IP tunnels, and create mirrored service ports on additional SIP interfaces.

Online Offline Configuration

The **set-system-state** ACLI command provides the ability to temporarily place a clustered SBC in the offline state. The offline setting puts the SBC into a state where it is powered on and available only for administrative purposes.

The transition to the offline state is graceful in that existing calls are not affected by the state transition. The SBC informs the SLB of the impending status change via a CCP message. Upon receiving such a message, the SLB ceases to forward new endpoints to the SBC, and places the SBC in the Shutdown state. The SBC, for its part, enters a state that results in the rejection of any incoming out-of-dialog SIP requests. Eventually all calls compete, registrations expire and are removed by the SLB, and returning endpoints are allotted to active SBCs.

Use the **set-system-state offline** ACLI command to place an SBC in the offline state.

```
ACMEPACKET# set-system-state offline
Are you sure you want to bring the system offline? [y/n]?: y
Setting system state to going-offline, process will complete when all current
calls have completed
ACMEPACKET#
```

Note: An SBC in the offline state plays no role in a balance or rebalance operation.

In a similar fashion use the **set-system-state online** ACLI command to place an SBC in the online state.

```
ACMEPACKET# set-system-state online
Are you sure you want to bring the system online? [y/n]?: y
```

```
Setting system state to online
ACMEPACKET#
```

4

SLB/Cluster Management & Diagnostics

SLB Statistics

The SLB provides the operator with a full set of statistical data for troubleshooting and diagnostic purposes. This section describes current statistical outputs and defines displayed values. It is important to become familiar with the data and the collection process when opening trouble tickets as service personnel will rely upon this information to assist you in diagnosing hardware, software, and/or network issues.

show balancer

The **show balancer** command is the root of all statistical data pertinent to SLB operation. Below is a list of valid arguments, which are described in further detail in the following sections:

```
ACMEPACKET# show balancer ?
end-points  show session load balancer end-points
members     show session load balancer cluster member summary
metrics     show load balancer metrics
realms      show load balancer realms
tunnels     show session load balancer statistics
statistics  show session load balancer IP-in-IP tunnel info
ACMEPACKET#
```

show balancer endpoints

The **show balancer endpoints** command displays a full list of all IP-to-SBC mappings resident in the SLB. As the SLB can hold up to ten million entries, the output of this command can and will grow very large, and extreme caution should be exercised when executing this command on a heavily trafficked SLB system.

```
ACMEPACKET# show balancer endpoints
IP address Port Access   Core     Flags    SBC  Handle
---------  ---- -------- -------- -------- ---- ------
15.0.0.24  5060 00134324 10134324 c0000000 1023 [wigglytuff@172.30.45.71]
15.0.0.22  5060 00134323 10134323 c0000000 1022 [jigglypuff@172.30.45.70]
15.0.0.20  5060 00134322 10134322 c0000000 1021 [tuono]
15.0.0.18  5060 00134321 10134321 c0000000 1020 [superduke]
15.0.0.16  5060 00134320 10134320 c0000000 1023 [wigglytuff@172.30.45.71]
15.0.0.14  5060 00134319 10134319 c0000000 1022 [jigglypuff@172.30.45.70]
15.0.0.12  5060 00134318 10134318 c0000000 1021 [tuono]
15.0.0.10  5060 00134317 10134317 c0000000 1020 [superduke]
15.0.0.8   5060 00134316 10134316 c0000000 1023 [wigglytuff@172.30.45.71]
```

```
15.0.0.6   5060 00134315 10134315 c0000000 1022 [jigglypuff@172.30.45.70]
15.0.0.4   5060 00134314 10134314 c0000000 1021 [tuono]
15.0.0.2   5060 00134313 10134313 c0000000 1020 [superduke]
ACMEPACKET#
```

The table provided by **show balancer endpoints** displays every endpoint mapping. In the above example, note that IP addresses in the 14.0.134.0/24 space are being distributed among a number of SLBs. The IP address and Port columns pinpoint a specific endpoint. The Index, Address, and Flags columns contain SLB internal reference identifiers for locating that specific endpoint in memory. The SLB Handle column identifies which SLB serves that endpoint; use the **show balancer members** command to display a mapping of SLB names to SLB handles.

You can use optional command arguments to filter/restrict command output.

show balancer endpoints <ip-address> restricts the display to one endpoint.

For example:

```
show balancer endpoints address 14.0.134.232
```

displays data for the specified IP endpoint

show balancer endpoints <ip-address>/<:port_num>restricts the display to a specific port on a specific IP address.

For example:

```
show balancer endpoints address 14.0.134.232:5060
```

displays data for port 5060 on the specified endpoint.

show balancer endpoints <ip-address>/<bit-mask-len> restricts the display to a contiguous range of endpoint addresses.

For example:

```
show balancer endpoints address 14.0.134.0/24
```

displays data for the 14.0.134.0 subnet.

```
show balancer endpoints address 14.0.134.240/28
```

displays data for endpoint addresses 14.0.134.240 through 14.0.134.255.

```
show balancer endpoints <ip-address>/<bit-mask-len><:port_num>
```

displays data for a specific port on a contiguous range of endpoint addresses.

show balancer members

The **show balancer members** command provides a list of all SBCs that have registered with the SLB.

```
ACMEPACKET# show balancer members
ural# show balancer members
SBC  Name                      Source Address   Destination Address  S/P/VLAN
Endpoints
---- ------------------------- ---------------- -------------------- --------
---------
1020 superduke                 68.68.68.100     68.68.68.5           0/0/0
3
1021 tuono                     68.68.68.100     68.68.68.4           0/0/0
3
1022 jigglypuff@172.30.45.70   68.68.68.100     68.68.68.1           0/0/0
3
1023 wigglytuff@172.30.45.71   68.68.68.100     68.68.68.2            0/0/0
3

              max endpoints: 12
    max untrusted endpoints: 200
```

```
          current endpoints: 12
current untrusted endpoints: 0
               current SBCs: 4
ACMEPACKET#
```

The SBC column contains the SBC handle, an internal shorthand that identifies a specific SBC. The **show balancer members** command provides a handle-to-hostname mapping.

Name contains the SBC hostname. Standalone SBCs are displayed as hostname@IP address, and highly available SBCs (csbc1a in the above display) are displayed as hostname.

Source IP contains the local (SLB) tunnel address.

Destination IP contains the remote (SBC) tunnel address.

Slot, Port, and Vlan identify the local interface that supports the SLB-to-SBC tunnel.

endpoints contains the number of endpoint-SBC associations that the SLB created for each specific SBC,

max endpoints contains the licensed capacity of the SBC.

max untrusted endpoints contains the maximum allowed number of untrusted endpoints.

current endpoints contains the current number of endpoints, trusted and untrusted

current untrusted endpoints contains the current number of untrusted endpoints.

show balancer metrics

The **show balancer metrics** command displays a comparison between the number of local endpoints (that is, the associations between source addresses and each SBC) and the number of remote endpoints (that is, what the SBC reports to the SLB as the number of endpoints it has received via the tunneled interface). Note that in the example output below those two numbers are the same; this is true if and only if there are no users in the access network that have multiple phone lines sourced from the same IP address. Were that the case, the number of remote endpoints would be higher than the number of local endpoints.

This table is populated with the data received in the periodic heartbeats from the SBC to the SLB. As these heartbeats are somewhat infrequent (every two seconds by default), the data in this table should only be considered accurate within two seconds.

```
ACMEPACKET# show balancer metrics
                              local  remote                     max
SLB Name                      epts   epts    max reg   CPU   CPU
--- ------------------------- ------ ------- -------- ----   ----
 93 magichat@172.30.68.34     0      0       480000    2.7  90.0
 94 westy                     0      0       480000    2.7  90.0
 95 samadams@172.30.68.33     0      0       480000    2.8  90.0
 96 bass@172.30.68.35         0      0       480000    4.3  90.0
 97 sixtus@172.30.68.36       0      0       480000    2.9  90.0
 98 newcastle@172.30.68.37    0      0       480000    2.9  90.0
 99 guiness@172.30.68.33      0      0       480000    3.6  90.0
ACMEPACKET#
```

SLB contains the SLB handle.

Name contains the SLB hostname.

max reg contains the maximum number of endpoints the SLB will send to this specific SBC. Its value is derived from the product of the **session-multiplier** parameter in the cluster-config configuration element and the SBC’s licensed session capacity. The SBC passes this value to the SLB during the SBC’s registration process into the cluster.

CPU contains the last received information on the CPU percentage from this SBC.

max CPU contains the threshold percentage at which the SBC is removed from consideration for the assignment of new endpoints. The default value is 90%, and may be changed on an SBC by setting the load-limit value as a SIP configuration option.

show balancer realms

The **show balancer realms** command displays a composite list of realms that all member SBCs have registered with the SLB.

```
ACMEPACKET# show balancer realms
Realm        SBC Tunnel Name                     ref count endpoints
----------- --- ------ ------------------------ --------- ---------
access       99   4092 newcastle@172.30.68.37           1     53535
access       98   4091 magichat@172.30.68.34            1     53535
access       97   4090 augustiner@172.30.68.41          1     53535
access       94   4086 bass@172.30.68.35                1     53535
access       93   4085 westy@172.30.68.42               1     53535
access       92   4084 sixtus@172.30.68.36              1     53535
access       96   4089 guiness@172.30.68.33             1     53535
net-13       99   4088 samadams@172.30.68.31            1     62550
net-13       91   4087 stbernie@172.30.68.43            1     62550
ACMEPACKET#
```

In this example, seven of the nine SBCs have registered the realm access and two have registered the realm net-13. The total number of endpoints for each of these services is indicated in the rightmost column. ref count is reserved for future use.

show balancer statistics

The **show balancer statistics** command displays statistical output pertinent to low-level events on the SLB. The contents and output of this command are subject to change, and will be documented in a subsequent document release.

```
ACMEPACKET# shower-balancer-statistics
LBP not initialized drops                  0
max capacity reached drops                 2
endpoint SBC mismatch errors              0
endpoint table read errors                0
Tx packet failed count                     0
service not found count                    0
duplicate ept packet drops                 0
msgq drops                                 0
forwarded duplicates                       0
policy miss                            40508
realm miss                                 0
throttle drops                             0
throttle skips                             0
throttle policy skips                      0
total packets processed                40508
endpoints removed                         0
EPT delete errors                          0
endpoints added                           0
EPT add errors                             0
EPT update errors                          0
group not found                            0
LBP agent not found                        0
invalid endpoint                          0
insert error                               0
untrusted dropped                          0
untrusted age outs                       171
packets dropped in balance                 0
packets dropped by standby                 0
-----------------------------
trusted endpoints (EPT db)                0
untrusted endpoints (EPT db)            337
-----------------------------
total trusted endpoints                   0
total untrusted endpoints               337
```

```
total endpoints                         337
ACMEPACKET#
```

show balancer tunnels

When implemented on the SLB, the **show balancer tunnels** command generates a list of data for each tunnel between the SLB and its clustered SBCs. It includes the tunnel source and destination addresses, as well as an internal switch ID (swid) for this tunnel.

```
ACMEPACKET# show balancer tunnels
1020(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.5
slot/port/vlan = 0/0/0
traffic policy selected: "" ;  traffic policy configured: implicit defaults.
  service: 172.16.2.3:5060 [access] protocols: 17/21588

1021(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.4
slot/port/vlan = 0/0/0
traffic policy selected: "" ;  traffic policy configured: implicit defaults.
  service: 172.16.2.3:5060 [access] protocols: 17/21588

1022(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.1
slot/port/vlan = 0/0/0
traffic policy selected: "" ;  traffic policy configured: implicit defaults.
  service: 172.16.2.3:5060 [access] protocols: 17/21588

1023(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.2
slot/port/vlan = 0/0/0
traffic policy selected: "" ;  traffic policy configured: implicit defaults.
  service: 172.16.2.3:5060 [access] protocols: 17/21588

ACMEPACKET# show balancer tunnels
errors       fragments    statistics
```

Use the **error** argument for error reporting and troubleshooting.

```
ACMEPACKET# show balancer tunnels errors
src addr 68.68.68.100 / dst addr 68.68.68.5 / slot 0 / port 0 / vlan 0:
  Proto Encaps Errors Decaps Errors
  ----- ------------- -------------
  17                0             0

src addr 68.68.68.100 / dst addr 68.68.68.4 / slot 0 / port 0 / vlan 0:
  Proto Encaps Errors Decaps Errors
  ----- ------------- -------------
  17                0             0

src addr 68.68.68.100 / dst addr 68.68.68.1 / slot 0 / port 0 / vlan 0:
  Proto Encaps Errors Decaps Errors
  ----- ------------- -------------
  17                0             0

src addr 68.68.68.100 / dst addr 68.68.68.2 / slot 0 / port 0 / vlan 0:
  Proto Encaps Errors Decaps Errors
  ----- ------------- -------------
  17                0             0

unknown protocol:      0
```

```
do not fragment drops: 0
no matching tunnel:    0
service lookup failed: 0
IP frag msg failure:   0
mblk alloc failues:    0
IP frame too large:    0
unknown errors:        0
unknown errors:        0
ACMEPACKET#show balancer tunnels
errors       fragments    statistics
```

The **show balancer tunnels error** command can also be executed on an SBC cluster member. In this usage, the displayed data is restricted to errors between the specific cluster member and the SLB.

Use the **fragments** argument for information related to packet fragmentation/reassembly details.

```
ACMEPACKET# show balancer tunnels fragments
src addr 68.68.68.100 / dst addr 68.68.68.5 / slot 0 / port 0 / vlan 0:
  IP:Port:      172.16.2.3:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  17             3         1239          0            0


src addr 68.68.68.100 / dst addr 68.68.68.4 / slot 0 / port 0 / vlan 0:
  IP:Port:      172.16.2.3:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  17             3         1240          0            0


src addr 68.68.68.100 / dst addr 68.68.68.1 / slot 0 / port 0 / vlan 0:
  IP:Port:      172.16.2.3:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  17             3         1243          0            0


src addr 68.68.68.100 / dst addr 68.68.68.2 / slot 0 / port 0 / vlan 0:
  IP:Port:      172.16.2.3:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  17             3         1244          0            0
ACMEPACKET#show balancer tunnels
errors       fragments    statistics
```

The **show balancer tunnels fragments** command can also be executed on an SBC cluster member. In this usage, the displayed data is restricted to fragmentation operations between the specific cluster member and the SLB.

Use the **statistics** argument for information related to packet counts.

```
ACMEPACKET# show balancer tunnels statistics
src ip 182.16.203.83 / dst ip 182.16.203.87 / slot 0 / port 1 / vlan 0:
  IP:Port:  192.169.203.83:5050
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  6              0            0          0            0
  IP:Port:  192.169.203.83:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
  ----- ---------- ------------ ---------- ------------
  17         48011     24213914          0            0
src ip 182.16.203.83 / dst ip 182.16.203.86 / slot 0 / port 1 / vlan 0:
  IP:Port:  192.169.203.83:5060
  Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
```

```
  ----- ---------- ------------ ---------- ------------
  17         48017     24217918          0            0
ACMEPACKET#
```

The **show balancer tunnels statistics** command can also be executed on an SBC cluster member. In this usage, the displayed data is restricted to traffic counts between the specific cluster member and the SLB.

Cluster Control Protocol Statistics

The CCP provides the operator with a full set of statistical data for troubleshooting and diagnostic purposes.

show ccd

The **show ccd** command is the root of all statistical data pertinent to CCP operation. Below is a list of valid arguments, which are described in further detail in the following sections:

```
ACMEPACKET# show ccd ?
ccp         Cluster Control Protocol Stats
rebalance   Display rebalance queue
reset       Reset Stats
sds         Controlled SDs
stats       Cluster Control Stats
ACMEPACKET#
```

show ccd ccp

The **show ccd ccp** command displays aggregated data (that is, from all cluster members) about specific CCP operations.

```
ACMEPACKET# show ccd ccp
--------------------
M01:0
--------------------
Svc Add                Recent      Total  PerMax
===================    ======  =========  ======
Ops Recvd                   0          8       4
Op Replies Sent             0          8       4
                       -------- Received -------   ----------  Sent --------
Status Code            Recent      Total  PerMax   Recent      Total  PerMax
-----------            ------  ---------  ------   ------  ---------  ------
200 OK                      0          0       0        0          8       4
EP Del                 Recent      Total  PerMax
===================    ======  =========  ======
Ops Recvd                   0       3984    1013
Duplicate Ops               0         12       6
Op Replies Sent             0       3984    1013
                       -------- Received -------   ----------  Sent --------
Status Code            Recent      Total  PerMax   Recent      Total  PerMax
-----------            ------  ---------  ------   ------  ---------  ------
200 OK                      0          0       0        0       3984    1013
EP Promo               Recent      Total  PerMax
===================    ======  =========  ======
Ops Recvd                   0     115601    8187
Duplicate Ops               0        329      23
Op Replies Sent             0     115601    8187
                       -------- Received -------   ----------  Sent --------
Status Code            Recent      Total  PerMax   Recent      Total  PerMax
-----------            ------  ---------  ------   ------  ---------  -
200 OK                      0          0       0        0     115601    8187
Metrics                Recent      Total  PerMax
===================    ======  =========  ======
Ops Recvd                  57      16330      75
```

```
Op Replies Sent           57      16331      75
                      -------- Received -------   ----------  Sent --------
Status Code           Recent      Total  PerMax   Recent      Total  PerMax
-----------           ------  ---------  ------   ------  ---------  -
200 OK                     0          0       0       57      16328      75
406 Not Accept             0          0       0        0          3       3
Prov Done             Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          6       3
Op Replies Sent            0          6       3
                      -------- Received -------   ----------  Sent --------
Status Code           Recent      Total  PerMax   Recent      Total  PerMax
-----------           ------  ---------  ------   ------  ---------  -
200 OK                     0          0       0        0          6       3
Going Down            Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          1       1
Op Replies Sent            0          1       1
                      -------- Received -------   ----------  Sent --------
Status Code           Recent      Total  PerMax   Recent      Total  PerMax
-----------           ------  ---------  ------   ------  ---------  -
200 OK                     0          0       0        0          1       1
Stop Down             Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          8       6
Op Replies Sent            0          9       6
                      -------- Received -------   ----------  Sent --------
Status Code           Recent      Total  PerMax   Recent      Total  PerMax
-----------           ------  ---------  ------   ------  ---------  -
200 OK                     0          0       0        0          6       3
406 Not Accept             0          0       0        0          3       3
ACMEPACKET#
```

Use the hostname argument to display data for a specific cluster member.

```
ACMEPACKET# show ccd ccp westy
--------------------
westy
--------------------
Svc Add               Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          1       1
Op Replies Sent            0          1       1
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 0          1       1
EP Del                Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          1       1
Op Replies Sent            0          1       1
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 0          1       1
EP Promo              Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0       1002     947
Op Replies Sent            0       1002     947
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 0       1002     947
Metrics               Recent      Total  PerMax
```

```
===================   ======  =========  ======
Ops Recvd                 20     167849      15
Op Replies Sent           20     167849      15
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 20     167849      15
Prov Done             Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          1       1
Op Replies Sent            0          1       1
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 0           1       1
Stop Down             Recent      Total  PerMax
===================   ======  =========  ======
Ops Recvd                  0          5       2
Op Replies Sent            0          5       2
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          0       0 0           5       2
ACMEPACKET#
```

show ccd sds

The **show ccd sds** command displays a table containing an overview of all of the data gleaned from the CCP from each SBC.

```
ACMEPACKET# show ccd sds
Session Director        Hdl  State     Tunnel Svcs Version     HW
----------------------- ---- --------- ------ ---- ---- ------ -----
augustiner@172.30.68.41   95 InService  2/2     2  6.2.0.30b8  SD3
bass@172.30.68.35         94 InService  2/2     2  6.2.0.30b8  SD3
guinness@172.30.68.33     96 InService  2/2     2  6.2.0.30b8  SD3
magichat@172.30.68.34     97 InService  2/2     2  6.2.0.30b8  SD3
newcastel@172.30.68.37    98 InService  2/2     2  6.2.0.30b8  SD3
samadams@172.30.68.31     99 InService  2/2     2  6.2.0.30b8  SD3
sixtus@172.30.68.36       92 InService  2/2     2  6.2.0.30b8  SD3
stbernie@172.30.68.43     91 InService  2/2     2  6.2.0.30b8  SD3
westy@172.30.68.42        93 InService  2/2     2  6.2.0.30b8  SD3
ACMEPACKET#
```

Session Director contains the hostname of the cluster SBCs that are connected to the SLB. As with all of the similar statistical output, standalone SBCs are displayed as hostname@eth0 IP address, and highly available SBCs are displayed as hostname.

Hdl contains the clustered SBC handle, an internal shorthand that identifies a specific cluster member. The **show balancer members** command provides a handle to hostname mapping.

State contains the current SBC state. Valid states are:

- Init — during initial handshaking with the SLB
- InService — healthy and operating normally
- Rebalance — during a cluster expansion/contraction operation
- LostControl — no longer communicating with the SLB

Tunnel contains the number of tunnels between the SBC and SLB.

Svcs contains the number of advertised services (protocols) that the SBC has negotiated with the SLB.

Version contains the software version running on that SBC.

HW identifies the hardware platform (in this case, SD3 identifies an Acme Packet 4500 SBC).

LastPing is not currently used.

When issued with an optional hostname argument, the **show ccd sds** command provides a detailed report for the target hostname.

```
ACMEPACKET# show ccd sds bass
Session Director: bass@172.30.68.35
+-------------------------------------------------------------------
|State     : InService       Handle       : 0x5f
|Tunnels   : 1               ServicePorts : 1
|HW Type   : SD3             SW Version   : 6.2.0.30b8
|Last Ping : 1080ms          App Count    : 1
|
|Service:         App  SvcPorts Tunnels endpoints DropCount
+---------------- ---- -------- ------- --------- ---------
access            SIP         1       1    285714         0
|
|#   Tunnel                            App  Handle     Svcs LastHB
+--- --------------------------------- ---- ---------- ---- -----
|0   (1.1.1.100|1.1.1.15)              SIP  0xffb         1  375ms
Traffic Policy: Implicit Defaults
|
|#   CPU   MAX   CurReg    RegLimit  CurSess   MaxSess
+--- ----- ----- --------- --------- --------- ---------
|0    4.1% 90.0%    285714         0      7336     64000
|     4.1% 90.0%    285714    960000      7736     64000
|
|Service Port                            App  Handle       TunNdx Avail
+--------------------------------------- ---- ---------- ------ -----
|access::192.168.168.100:5060<17>       H248     513(1)      0   yes
ACMEPACKET#
```

SBC State

- State — the current SBC state
- Handle — the SBC handle
- Tunnels — the current number of SBC tunnels
- ServicePorts — the current number of SBC service ports
- HW Type — the hardware platform (in this case, SD3 identifies an Acme Packet 4500 SBC)
- SW Version — the installed software revision level
- Last Ping — the number of elapsed milliseconds, since a ping/keepalive was received from this SBC
- App Count — the number of applications supported by the SBC

Services State

- Service — the realm advertised by the SBC in the Service Port ID
- App — the supported protocol: SIP
- SVCPorts — the current number of service ports
- Tunnels — the current number of tunnels
- endpoints — the cumulative number of endpoints for this service
- DropCount — the number of elements to drop when rebalancing this SBC

Tunnel State

- # — the tunnel index (0 or 1)
- Tunnel — the SLB and SBC tunnel IP address
- App — the supported protocol: SIP
- Handle — the handle for the tunnel
- Svcs — the number if service ports supporting the tunnel
- LastHB — the number of elapsed milliseconds since a heartbeat was received from the remote end of this tunnel

Tunnel Metrics

- # — the tunnel number (0 or 1)
- CPU — the current CPU utilization rate
- Max — the maximum supported CPU utilization rate, if this value is exceeded, the tunnel implements a load limit algorithm
- CurReg — the current number of registrations supported by the SBC
- regLimit — the maximum number of registrations supported by the SBC
- CurSess — the current call count reported by the SBC
- MaxSess — the maximum sessions for which the SBC is licensed

Service Port Data

- Service Port — the service path (the concatenation of realm, IP address, port number, and IP Level 4 protocol number — 17 for UDP, 6 for TCP)
- App — the supported protocol: SIP
- Handle — the handle for the service port
- TunNdx — the tunnel the service port is registered for
- Avail — current availability (yes or no) determined by the presence of heartbeats

show ccd stats

The **show ccd stats** command displays endpoint statistics for the SBC members of the cluster.

```
ACMEPACKET# show ccd stats
17:10:09-54
                          -------- Period -------    ---- LifeTime ----
 SD                        Active  Rate   High Total   Total PerMax    High
bass@172.30.68.35        I285714   0.0 285714    0 285.71K 13.76K 285.71K
guinness@172.30.68.33    I285714   0.0 285714    0 285.71K 13.76K 285.71K
magichat@172.30.68.34    I285714   0.0 285714    0 285.71K 13.76K 285.71K
newcastel@172.30.68.37   I285714   0.0 285714    0 285.71K 13.76K 285.71K
samadams@172.30.68.31    I285714   0.0 285714    0 285.71K 13.76K 285.71K
sixtus@172.30.68.36      I285714   0.0 285714    0 285.71K 13.76K 285.71K
westy                    I285714   0.0 285714    0 285.71K 13.76K 285.71K
Total endpoints: 153908
Total SDs      : 9
ACMEPACKET#
```

The Period stats provided represent an accumulation of data for the amount of time specified after the dash separator in the timestamp printed in the first line of output (in this example, the period represents 54 seconds).

The single ASCII character between the SD column and the Active column is the state of that SBC; the letter I represents InService.

The Rate column displays the transmission rate of new endpoint associations to that particular SBC. (In the sample, no new endpoints are arriving in the cluster, so all of the SBCs show a rate of 0.0.) The High field indicates the highest number of active endpoint associations for the current period.

When issued with an optional hostname argument, the **show ccd stats** command provides a detailed report for the target hostname.

```
ACMEPACKET# show ccd stats bass
15:09:25-59
SD bass@172.30.68.33 [InService]
State                 -- Period -- -------- Lifetime --------
            Active    High   Total      Total Permax     High
Tunnels          1       1       0          2      1        1
Service Ports    2       2       0          2      1        2
endpoints    53571   53571      0      53571  14399    53571
Contacts      53571   53571      0      53571  14399    53571
```

```
Sessions         0       0       0          0      0        0
Remote CPU       0       0       0          0      0        0
SD Something        ---- Lifetime ----
                          Recent        Total       PerMax
Heartbeats rcvd              30       27426           15
Heartbeats Missed             0           1            1
Tunnel Adds                   0           2            1
Tunnel Removes                0           1            1
Service Adds                  0           2            1
Service Removes               0           0            0
endpoint Removes             0           0            0
endpoint Promotes            0       53571        13561
endpoints Skipped            0           0            0
Rebalance Source              0           0            0
Rebalance Targe               0           0            0
Rebalance Request             0           0            0
Rebalance Replies             0           0            0
CPU Above Limit               0           0            0
CPU Above Threshold           0           0            0
Online Transitions            0           0            0
Offline transitions           0           0            0
Tunnel Add Fails              0           0            0
CCD Tunnel Add Fails          0           0            0
Tunnel Remove Fails           0           0            0
CCD Tunnel Remove Fails       0           0            0
Service Add Fails             0           0            0
CCD Svc Add Fails             0           0            0
Service Remove Fails          0           0            0
CCD Svc Remove Fails          0           0            0
Service Adds No Cfg           0           0            0
Bad Service Handle            0           0            0
endpoint Remove Fails        0           0            0
endpoint Prom Fail           0           0            0
ACMEPACKET#
```

The **Period** stats provided represent an accumulation of data for the amount of time specified after the dash separator in the timestamp printed in the first line of output (in this example, the period represents 59 seconds).

Tunnels contains the number of tunnels between the SLB and the target SBC, in this case, bass.

Service Ports contains the number of Service Ports advertised by the target SBC when it joined the cluster.

endpoints and **Contacts** contain the number of endpoint associations the SLB has assigned to the target SBC. If there is only one registering device at a given endpoint, a one-to-one correlation between endpoints and contacts is expected. However, if the **atom-limit-divisor** parameter has been set to a non-default value, the number of contacts exceeds the number of endpoints.

Sessions contains the number of active calls.

The table below the overview data displays specific CCP message statistics.

HeartBeats rcvd contains the number of heartbeat/keepalive messages received from the target SBC. Heartbeats are sent every two seconds by the SBC.

HeartBeats Missed contains the number of scheduled heartbeat/keepalive messages not received from the target SBC.

The **Tunnel Adds** and **Tunnel Removes** counters are incremented when an SBC joins the cluster and leaves the cluster, respectively.

The **Service Adds** and **Service Removes** counters are incremented when an SBC advertises support for a service and withdraws support for a service, respectively. This generally happens only when an SBC first joins the cluster, or if the configuration on a clustered SBC is changed, saved, and activated.

The **endpoint Removes** counter tracks the number of SBC-originated Cluster Control messages that request the SLB to delete a forwarding rule. Such a request can be the result of (1) a rebalance operation (when the SLB asks for the SBC to nominate candidates for rebalancing), (2) an endpoint de-registration with the SBC, or (3) an endpoint is power down. Generally, whenever a registration cache entry on a clustered endpoint is removed by the SBC, it notifies the SLB to remove that binding.

The **endpoint Promotes** counter tracks the number of promotion messages the SBC sends to the SLB to validate an untrusted forwarding rule. When the SLB first creates a forwarding rule for a new endpoint, it treats it as untrusted. When the SBC receives a 200 OK for a REGISTER message from that endpoint's registrar, the SBC sends a Promote Cluster Control message to the SLB. At this point, the SLB modifies the particular forwarding rule and assigns it trusted status. If this Promote message is not received within the time configured as the untrusted-grace-time in the lbp-config, the SLB deletes the untrusted entry.

endpoints Skipped contains the number of endpoints in its registration cache that the SBC has skipped over during a rebalance request. Skipping may be done for one of two reasons: either the most appropriate user for rebalancing was in an active phone call (and **rebalance-skip-calls** was enabled in cluster-config), or the **rebalance-skip-ahead** value in cluster-config was set to a nonzero value. In this case, when the SBC is asked to nominate users for rebalance, it will skip over any users whose registration cache entry is due to expire within the number of milliseconds set as the **rebalance-skip-ahead** value.

Rebalance Source contains the number of times the target SBC was used as a source of endpoints during a rebalance operation (that is, it supplied endpoints to a cluster member that was added to the cluster after itself).

Rebalance Target contains the opposite: the number of times that SBC was the recipient of endpoints from other sources during a rebalance operation.

The **Rebalance Requests** and **Rebalance Replies** counters increment upon receipt of a Cluster Control message from the SLB to the SBC asking it to divest itself of endpoints, and the responsive Cluster Control message from the SBC that indicates the endpoints the SBC has chosen.

The **CPU Above Limit** and **CPU Above Threshold** counters increment whenever an SBC has reported a high CPU value, and has been taken out of consideration for new endpoint assignments. Generally, the CPU limit and threshold are the same value (90%). However, it is possible to configure the threshold to be lower using the sip-config **option load-limit**.

SBC Cluster Member Statistics

The SBC cluster member also provides the operator with summary statistical data for active endpoints

show sip lb-endpoints

The **show sipd lb-endpoints** command displays SBC endpoint stats by realm or tunneled service ports, by sip-interface since each SIP interface is uniquely identified by its realm name.

While this command was not changed for the addition of source port keys, there are some important items to note. When all endpoints are behind a NAT and source ports are used in endpoint keys, the number of endpoints should match the number of atoms. Were all endpoints are behind a single NAT, and source address keys in use, There would be many atoms and only one endpoint. Obviously in mixed environments this will be less clear and thus this command less useful. However, in lab environments this can be useful.

S-Cz7.2.0M6

```
ACMEPACKET# sho sipd lb-endpoints
-------------------------------------
Realm Endpoint Stats
-------------------------------------
10:57:29-35
Service Realm192p1
```

```
               ------- Period -------       ------ Lifetime -----
               Active    High   Total       Total  PerMax    High
Endpoints      2                            2
2                      2                  2                           2
Atoms          2                            2
2                      2                  2                           2
                         ------- Lifetime -------
                         Recent      Total  PerMax
Refreshes                   0          0       0
Adds                        2          2       2
Low Skips                   0          0       0
High Skips                  0          0       0
Auth Promo Tries            0          0       0
noTrust Promo Tries         0          0       0
Promo Tries                 0          0       0
Remove Conflicts            0          0       0
Remote Deletes              0          0       0
SP Removes                  0          0       0
Expiry Deletes                       0          0       0
Session Deletes                0          0       0
Session Adds                   0          0       0
Move Deletes                   0          0       0
Move Del No Tells              0          0       0
SvcMove Deletes                0          0       0
SvcMove Del NoTell             0          0       0
Auth Promotes                  0
0                      0
Auth Deletes                   0          0       0
Add Errors                     0          0       0
Delete Deny Sess               0          0       0
Delete Deny Reg                0          0       0
Delete Deny Purge              0          0       0
Delete Missing                 0          0       0
Delete Errors                  0          0       0
Update Deny Purge              0          0       0
Auth Deny Purge                0          0       0
Remote Sess Skips              0          0       0
Remote Del Fails               0          0       0
SP Remove Fails                0          0       0
Expiry Del Fails               0          0       0
Sess Del Fails                 0          0       0
Sess Add Fails                 0          0       0
Move Del Fails                 0          0       0
Move No Tell Fails             0          0       0
SvcMove Del Fails              0          0       0
SvcMv NoTell Fails             0          0       0
Auth Promo Fails               0          0       0
Auth Del Fails                 0          0       0
App Cache Dels                 0          0       0
```

show sip ccp

The **show sip ccp** command displays a cluster-member-specific summary of CCP operations.

```
westy# show sip ccp
--------------------
M00:0.4:T2
--------------------

EP Del                Recent      Total  PerMax
===================   ======  =========  ======
Ops Sent                   0          1       1
Op Replies Recvd           0          1       1
```

```
                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          1       1 0           0       0

EP Promo              Recent      Total  PerMax
===================   ======  =========  ======
Ops Sent                   0        992     538
Op Replies Recvd           0        992     538

                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0        992     538 0           0       0

Metrics               Recent      Total  PerMax
===================   ======  =========  ======
Ops Sent                  25        207      15
Op Replies Recvd          25        207      15

                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                    25        207      15 0           0       0

Stop Down             Recent      Total  PerMax
===================   ======  =========  ======
Ops Sent                   0          2       2
Op Replies Recvd           0          2       2

                      -------- Received ------- ---------- Sent --------
Status Code           Recent      Total  PerMax Recent      Total  PerMax
-----------           ------  ---------  ------   ------ ---------  ------
200 OK                     0          2       2 0           0       0
westy#
```

5

Subscriber-Aware Load Balancer SNMP Reference

Overview

This chapter provides an overview of SNMP support for Subscriber-Aware Load Balancer (SLB) features.

Enterprise Traps

The following table identifies the SLB proprietary traps supported by the SLB.

apSLBEndpointCapacityThresholdTrap	Generated when the number of endpoints on the SLB exceeds the configured threshold.
apSLBEndpointCapacityThresholdClearTrap	Generated when the number of endpoints on the SLB falls below the configured threshold.
apSLBUntrustedEndpointCapacityThresholdTrap	Generated when the number of untrusted endpoints on the SLB exceeds the configured threshold.
apSLBUntrustedEndpointCapacityThresholdClearTrap	Generated when the number of untrusted endpoints on the SLB falls below the configured threshold.

License MIB (ap-license.mib)

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apLicenseEntry (1.3.6.1.4.1.9148.3.5.1.1.1)		
apLicenseSLBEndpointCap	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.23	SLB endpoint capacity

Subscriber-Aware Load Balancer MIB (ap-slb.mib)

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apSLBMIBObjects (1.3.6.1.4.1.9148.3.11.1)		
Object Identifier Name: apSLBMIBGeneralObjects (1.3.6.1.4.1.9148.3.11.1.1)		
apSLBStatsEndpointsCurrent	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.1	Number of endpoints currently on the SLB.
apSLBStatsEndpointsDenied	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.2	Number of endpoints denied by the SLB because the system has reached the maximum endpoint capacity.
apSLBEndpointCapacity	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.3	Maximum number of endpoints allowed on the SLB. This value is based on the installed SLB license(s).
apSLBEndpointCapacityUpperThresh	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.4	Percentage of endpoints relative to maximum threshold capacity.
apSLBEndpointCapacityLowerThresh	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.5	Percentage of endpoints relative to minimum threshold capacity.
apSLBStatsUntrustedEndpointsCurrent	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.6	Number of untrusted endpoints currently on the SLB.
apSLBStatsTrustedEndpointsCurrent	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.7	Number of trusted endpoints currently on the SLB.
apSLBStatsUntrustedEndpointsDenied	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.8	The number of untrusted endpoints denied by the SLB due to the total number of untrusted endpoints exceeding the configured maximum threshold.
apSLBStatsUntrustedEndpointsAgedOut	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.9	The number of untrusted endpoints aged out of the system because they were not authenticated within the configured grace period.
apSLBUntrustedEndpointCapacity	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.10	Maximum number of untrusted endpoints allowed on the SLB. This value is a configured percentage of the maximum endpoint capacity of the system.
apSLBUntrustedEndpointCapacityUpperThresh	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.11	Percentage of untrusted endpoint maximum threshold capacity in use.
apSLBUntrustedEndpointCapacityLowerThresh	apSLBMIBGeneralObjects: 1.3.6.1.4.1.9148.3.11.1.1.12	Percentage of untrusted enpoint minumum threshold capacity percentage.