

**Oracle® Communications Session
Director**

ACLI Configuration Guide

Release S-D7.1.0

Formerly Net-Net Session Border Controller

December 2014

Copyright ©2014 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Contents iii

About this Guide	xxix
Overview	xxix
Associated Documentation	xxix
Revision History	xxx
 1 Getting Started	 33
Introduction	33
Installation and Start-Up	33
Hardware Installation Summary	33
Connecting to Your Net-Net System	34
System Boot	35
ACLI System Prompt	36
Net-Net System Boot Parameters	37
Displaying Boot Parameters	37
Changing Boot Parameters	39
Your Net-Net System Image	41
Obtaining a New Image	41
Using FTP to Copy an Image on Your Net-Net System	41
Booting an Image on Your Net-Net 9200	43
Booting from Flash Memory	43
Booting from an External Device	44
Setting Up System Basics	44
NTP Synchronization	45
Setting the Time Zone	46
Virtual Management IP Address	46
Software Licensing	47
Supported Components	48

Unlicensed Net-Net Systems	49
Obtaining a License	49
Adding Licenses	50
Deleting Licenses	51
Temporary License Expiration	51
Viewing Licenses	51
RADIUS Authentication	52
How It Works	52
PAP Handshake	53
CHAP Handshake	54
MS-CHAP-v2 Handshake	54
Management Protocol Behavior	55
Local Authorization and RADIUS Authentication	56
ACLI Instructions and Examples	56
2 System Configuration	59
Introduction	59
General System Information	59
ACLI Instructions and Examples	59
Physical Interfaces	60
Overview	60
Types of Physical Interfaces	60
Slot and Card Location	60
NIU GigE Utilization	61
Management Interfaces	62
Before You Configure	63
ACLI Instructions and Examples	63
Network Interfaces	64
Overview	64
IP Configuration	65
Configuring Network Interfaces	66
ACLI Instructions and Examples	66
Secondary Gateway	68
SNMP	70
Overview	70
Configuring SNMP	70
SNMP Configuration Overview	71
ACLI Instructions and Examples	71
Syslog and Process Logs	74
Overview of Log Files	74

ACLI Instructions and Examples	74
Host Routes	78
Overview	78
ACLI Instructions and Examples	78
Source-based Routing	79
ACLI Instructions and Examples	79
System Access Control Lists	79
ACLI Instructions and Examples	80
Notes on Deleting System ACLs	80
System TCP Keepalive Settings	81
Trap Event Monitoring	82
ACLI Instructions and Examples	83
Accounting	83
SIP Session Accounting	83
ACLI Instructions and Examples	84
3 Realms and Nested Realms	89
Introduction	89
Overview	89
Configuring Realms	90
Before You Configure	90
ACLI Instructions and Examples	91
Nested Realms	94
Configuring Nested Realms	96
ACLI Instructions and Examples	97
Aggregate Session Constraints: Nested Realms	97
Realm-Based Packet Marking	99
About TOS/DiffServ	99
Packet Marking for Media	100
Configuring Packet Marking by Media Type	100
Packet Marking for Signaling	101
ACLI Instructions and Examples	101
SIP-SDP DCSP Marking/ToS Bit Manipulation	103
ACLI Instructions and Examples	104
Steering Pools	105
Configuration Overview	105
ACLI Instructions and Examples	106
Multiple Interface Realms	106

How It Works	107
Steering Pool Port Allocation	108
ACLI Instructions and Examples.....	109
Media over TCP	110
TCP Bearer Conditions.....	110
TCP Port Selection	111
SDP Offer Example.....	113
Timers	114
ACLI Instructions and Examples.....	114
Media Release Behind the Same IP Address	115
Additional Media Management Options	115
Configuring Media Release Behind the Same IP Address	115
Bandwidth CAC for Media Release	116
ACLI Instructions and Examples.....	116
Media Release between Endpoints with the Same IP Address	116
ACLI Instructions and Examples.....	117
Media Release Behind the Same NAT IP Address	117
ACLI Instructions and Examples.....	118
Media Profiles Per Realm	118
Call Admission Control and Policing	118
ACLI Instructions and Examples.....	119
Peer to Peer MSRP TCP Stitching.....	120
ACLI Instructions and Examples.....	121
 4 SIP Signaling Services.....	 123
Introduction.....	123
About the Net-Net 9200 and SIP.....	123
Types of SIP Devices.....	123
Basic Service Models.....	124
About SIP Interfaces.....	125
SIP Network Models and Types	125
Network Model.....	125
Network Type	126
Default Network Model and Type.....	126
Upgrade Considerations	126
ACLI Instructions and Examples.....	126
The Net-Net 9200 and SIP Signaling	127
The Home Realm	128

Overview	128
ACLI Instructions and Examples.	129
Setting Up a SIP Interface.	130
Overview	130
About SIP Ports.	130
Proxy Mode	131
Redirect Action	131
SIP maddr Resolution.	131
Trust Mode	133
ACLI Instructions and Examples.	134
SIP Timers	140
Overview	140
ACLI Instructions and Examples.	140
Configuring Timers for a sip-interface	142
ACLI Instructions and Examples.	142
SIP Options Tag Handling	143
Overview	143
Configuration Overview.	143
ACLI Instructions and Examples.	144
SIP Options	146
Overview	146
Global SIP Options.	146
SIP Interface Options	151
SIP Session Agent Options	152
SIP Realm Options	152
SIP Security	152
Denial of Service Protection.	152
Configuration Overview.	153
Configuring Security.	154
The SIP NAT Function	154
Overview	154
About Headers	156
SIP NAT Function Cookies	157
Configuration Overview.	159
ACLI Instructions and Examples.	161
SIP Realm Bridging	165
About SIP NAT Bridging	165
SIP NAT Bridge Configuration Scenarios.	166
SIP NAT Bridge Configuration	167
Shared Session Agent.	169

SIP Hosted NAT Traversal (HNT)	170
About HNT	170
Working with Multiple Domains.	173
TCP NAT Interval	174
HNT Configuration Overview	174
ACLI Instructions and Examples.	175
HNT Forced Unregistration	178
When to Use Forced Unregistration	179
Caution for Using Forced Unregistration	179
Configuring Forced Unregistration	180
Adaptive HNT	180
Overview	180
Adaptive HNT Example	181
Synchronize A-HNT Successful Timer to Standby	181
ACLI Instructions and Examples.	181
SIP Server Redundancy	182
Overview	182
Configuration Overview.	183
ACLI Instructions and Examples.	184
SIP Distributed Media Release (DMR)	184
Overview	184
Overview of SIP DMR Configuration	186
ACLI Instructions and Examples.	187
Embedded Header (E911) Support	188
ACLI Instructions and Examples.	189
Static SIP Header and Parameter Manipulation	189
Header Manipulation Rules.	190
Header Element Rules	190
About SIP Header and Parameter Manipulation.	190
Configuring SIP Header and Parameter Manipulation	190
SIP HMR (Header Manipulation Rules)	195
How It Works	196
Performing HMR on a Specific Header	197
MIME Body Manipulation Support.	199
Best Practices.	202
About Regular Expressions	203
Multiple SIP HMR Sets	204
Boolean logic within Content-Type	206
ACLI Instructions and Examples.	206
Configuration Examples.	209

SIP HMR: Response Manipulation	222
ACLI Instructions and Examples	222
Sample Configuration	224
HMR Functional Enhancements	225
Parenthesis Support	225
ACLI-based Syntax Checking	226
Escape Character Support	226
Boolean Comparison Type	226
Example Message Headers	227
Header Instance Selection	228
Stored Value Case Discrimination	230
Dialog-Matching Header Manipulation	230
About Dialog-Matching Header Manipulations	231
Built-In SIP Manipulations	233
Testing SIP Manipulations	234
Unique HMR Regex Patterns	234
Manipulation Pattern Per Remote Entity	234
Reject Action	235
Log Action	238
Changes to Storing Pattern Rule Values	238
Removal of Restrictions	238
Name Restrictions for Manipulation Rules	239
New Value Restrictions	239
HMR For SIP-ISUP	239
Dialog Transparency	239
Overview	239
SIP Via Transparency	240
How it Works	240
ACLI Instructions and Examples	241
Full Via Header Replacement	241
ACLI Instructions and Examples	242
Symmetric Latching	242
ACLI Instructions and Examples	242
Configuring RTCP Symmetric Latching	243
SIP Number Normalization	243
How it Works	243
ACLI Instructions and Examples	245
SIP Port Mapping	246
About SIP Port Mapping	246

How SIP Port Mapping Works	247
About NAT Table ACL Entries	248
Configuring SIP Port Mapping	249
SIP Registration Overload Protection	251
How It Works	251
ACLI Instructions and Examples	251
SIP Request Method Throttling	252
How It Works	252
ACLI Instructions and Examples	253
Requirements	253
SIP Delayed Media Update	256
Delayed Media Update Disabled	256
Delayed Media Update Enabled	257
ACLI Instructions and Examples	257
SIPConnect	258
Modifications to Registration Caching Behavior	258
Configuring SIP Connect Support	259
ACLI Instructions and Examples	259
SIP Registration Burst Rate Control	260
ACLI Instructions and Examples	261
SIP Mid-Call Re-Registration Following Service Route	261
How It Works	262
ACLI Instructions and Examples	262
SIP Session Agent Group Recursion	263
How It Works	263
ACLI Instructions and Examples	263
Configurable Route Recursion	264
Example 1	265
Example 2	265
ACLI Instructions and Examples	266
Event Package Interoperability	267
ACLI Instructions and Examples	267
SIP REGISTER Forwarding After Call-ID Change	268
Instructions and Examples	268
SIP Local Response Code Mapping	269
Instructions and Examples	269
Session Agent Ping Message Formatting	271
Instructions and Examples	271

SIP PPI—PAI Conversion	272
Instructions and Examples	272
Configuring SIP Statuses to Q.850 Reasons	272
SIP-SIP Calls	273
Trunk Group URIs	277
Terminology	277
Trunk Group URI Parameters	278
How It Works	281
ACLI Instructions and Examples	282
SIP Early Media Suppression	287
How It Works	288
About the Early Media Suppression Rule	290
Selective Early Media Suppression	290
Configuring Early Media Suppression	291
SIP SDP Insertion for Re-INVITES	294
How It Works	294
Restricted Media Latching	297
About Latching	297
How it Works	298
ACLI Instructions and Examples	299
Enhanced SIP Port Mapping	300
Anonymous Requests	300
SIP Registration Via Proxy	301
Considerations for Reg-Via-Key and Port Mapping	301
Request Routing	301
SIP Registration Cache Limiting	302
How It Works	302
About Registration Cache Additions, Modifications, and Removals	302
Registration Cache Alarm Threshold	303
Notes on Surrogate Registration	303
SIP Registration Event Package Support	303
Updating Expiration Values	304
ACLI Instructions and Examples	304
SIP Method-Transaction Statistic Enhancements	305
How It Works	305
ACLI Instructions and Examples	306
National Security and Emergency Preparedness for SIP	306
How It Works	306

ACLI Instructions and Examples	309
NSEP Enhancements	315
NSEP for Session Agents	315
Call Treatment.	315
Log Level Changes	316
SIP TCP Connection Reuse	316
How It Works	316
ACLI Instructions and Examples.	317
SIP Enforcement Profile and Allowed Methods	317
Monitoring.	319
Local Policy Session Agent Matching for SIP	320
How It Works	320
ACLI Instructions and Examples.	323
SIP Per User Subscribe Dialog Limit	324
Limit Enforcement	324
About Wildcarding	324
ACLI Instructions and Examples.	325
STUN Server	326
About STUN Messaging	327
STUN Server Functions on the Net-Net SBC	328
RFC 3489 Procedures	328
ACLI Instructions and Examples.	329
SIP Message Size Limiting	330
ACLI Configurations and Instructions	330
SIP SDP Address Correlation	331
ACLI Instructions and Examples.	331
SIP Unauthorized Endpoint Call Routing	332
SIP Registration Failure Mapping.	333
SIP SDP Anonymization	334
ACLI Instructions and Examples.	334
SIP Accounting	335
SIP Session Accounting	335
SIP CDR Generating Behaviors.	336
ACLI Instructions	337
SIP GRUU	337
Contact Header URI Replacement	338
Record-Route Addition	338
GRUU URI Parameter Name.	338

ACLI Instructions and Examples	339
SIP Proxy Mode High Subscription-Dialog Capacity	339
Subscription Requests	339
Topology Hiding	340
ACLI Instructions	340
SIP Diversion to SIP-ISUP Interworking	340
ACLI Instructions and Examples	340
SIP-ISUP Format Version Interworking	342
Details	342
ACLI Instructions and Examples	343
HMR for SIP-ISUP	344
Changes and Additions to Equality Operators	344
Reserved Words	345
Changes to Action	346
About MIME Rules	346
About MIME ISUP Manipulation	348
ACLI Instructions and Examples	352
Configuration Example	354
Expedited Call Leg Release for Preempted Hairpin Calls	355
Accounting Considerations	355
Expedited Media Termination of Calls with Abnormal Signaling Termination	356
Accounting Considerations	356
ACLI Instructions	356
5 H.323 Signaling Services	359
Introduction	359
Peering Environment for H.323	360
Overview	361
Signaling Modes of Operation	361
Realm Bridging with Static and Dynamic Routing	364
Before You Configure	364
Configuring Global H.323 Settings	364
ACLI Instructions and Examples	364
Configuring H.323 Interfaces	366
ACLI Instructions and Examples	366
Configuring H.323 Service Modes	368
ACLI Instructions and Examples	368
H.323 Features	370

Fast Start/Slow Start Translations	370
H.323 Registration Proxy	374
H.323 Registration Caching	378
H.245 Stage	383
Dynamic H.245 Stage Support	383
H.323 HNT	385
H.323 Party Number-E.164 Support	387
Signaling Only Operation	387
Maintenance Proxy Function	389
Automatic Gatekeeper Discovery	390
H.323 Alternate Routing	390
H.323 LRQ Alternate Routing	393
H.323 CAC Release Mechanism	395
H.323 Bearer-Independent Setup	396
TOS Marking for H.323 Signaling	398
H.323 Codec Fallback	398
H.323/TCS Media Sample Size Preservation	401
H.323-TCS: H.245 Support for H.264 and G722.1	403
International Peering with IWF and H.323 Calls	405
ACLI Instructions and Examples	405
Options	406
H.323 Automatic Features	409
Alias Mapping	409
Call Hold and Transfer	409
Media Release for SS-FS Calls	415
H.323 and IWF Call Forwarding	417
H.323 NOTIFY Support	419
ACLI Signaling Mode Configuration Examples	419
Configuration Fields and Values for B2BGW Signaling	420
Back-to-Back Gatekeeper Proxy and Gateway	422
Interworking Gatekeeper-Gateway	424
Additional Information	426
About Payload Types	426
H.323 Release Complete Cause Code Customization	430
H.323 Switchover Behavior	431
STOP CDRs	431
Graceful Media Stream Termination	431
H.323 Switchover Configuration	431
Cause Code Completeness	432
H.323 CDR Generating Behaviors	432
ACLI Instructions	433

H.323 Generating Behavior for OOS Session Agents	434
ACLI Instructions	434
Expedited Call Leg Release for Preempted Hairpin Calls	435
6 IWF Services	437
Introduction.	437
Access Network Application	437
Networking Peering Application.	438
How It Works	439
SIP/H.323 Negotiation: H.323 Fast Start	439
SIP/H.323 Negotiation: H.323 Slow Start.	441
Status and Codec Mapping	443
IWF RAS Registration Failure Code Mapping	444
IWF Service Enhancements	448
SIP Redirect—H.323 LRQ Management	448
SIP INFO and DTMF UII Management	450
Mid-Session Media Change	451
Early Media	451
Display Name Mapping.	451
IWF Ringback Support	451
H.323 Endpoint-Originated Call Hold and Transfer.	458
Music On Hold	461
Conference	464
IWF Call Forwarding	465
Media Release for H.323 SS-FS Calls for IWF	466
Before You Configure	469
H.323 Configuration.	469
SIP Configuration	469
The Role of Local Policy.	470
Configuring Interworking	471
ACLI Instructions and Examples.	471
IWF Inband Tone Option	472
ACLI Instructions and Examples.	472
RFC 3326 Support.	473
Default Mappings	474
ACLI Instructions and Examples.	476
IWF Privacy: Caller Privacy on Unsecure Networks.	477
About the Presentation Indicator	478
H.323 to SIP IWF Call.	478

SIP to H.323	479
IWF Privacy: Caller Privacy on Secure Connections.....	481
H.323 to SIP IWF	482
H.323 to SIP	482
SIP to H.323	483
IWF Privacy Extensions for Asserted Identity in Untrusted Networks.....	485
IWF Call Originating in H.323.....	486
Before You Configure	487
ACLI Instructions and Examples.....	487
IWF Privacy for Business Trunking	488
A Call Originating in H.323	488
A Call Originating in SIP	490
ACLI Instructions and Examples.....	492
Trunk Group URIs	493
Terminology	494
Trunk Group URI Parameters	494
How It Works	497
ACLI Instructions and Examples.....	499
IWF COLP/COLR Support.....	504
How It Works	505
ACLI Instructions and Examples.....	506
Options for Calls that Require the IWF.....	508
Suppress SIP Reliable Response Support for IWF	509
ACLI Instructions and Examples.....	510
IWF Codec Negotiation: H.323 Slow Start to SIP.....	510
ACLI Instructions and Examples.....	510
IWF: H.245 Signaling Support for G.726	511
How It Works: IWF.....	511
ACLI Instructions and Examples.....	511
International Peering with IWF and H.323 Calls	513
ACLI Instructions and Examples.....	513
7 IMS Support	515
Net-Net 9200 IMS Support.....	515
IMS Access Border Functions	516
P-CSCF Functions	516
SBC Protecting the P-CSCF.....	516
A-BGF Functions	516

IMS Interconnect Border Functions	517
Interconnect Border Control Function (I-BCF)	517
Interconnect-Border Gateway Function (I-BGF)	517
IMS Path and Service Route Header Support	518
Path Header	518
Service Route Header	518
Summary	519
Service Route Enhancements	520
SIP Interface: Implicit Service Route	520
Service Route Use	521
Notes About Upgrading	522
P-CSCF Endpoint Identification Using Address and Port	522
ACLI Instructions and Examples	522
IMS Support for Private Header Extensions for 3GPP	523
P-Associated-URI Header	523
P-Asserted-Identity Header	523
P-Called-Party-ID Header	525
P-Visited-Network-ID Header	525
P-Asserted-ID Output to CDRs	526
ACLI Configuration	526
IMS Charging Headers	526
P-Charging-Function-Address	528
Missing SIP Interface	529
RADIUS Accounting of Charging Headers	529
IP-PBX Surrogate Registration	530
Integrating with IMS	530
How it Works	531
Routing Calls from the IMS Core	532
Registration Retry Mechanism	532
Automatic Increment Mechanism	532
Routing Calls from the IP-PBX	533
Configuring Surrogate Registration	533
Viewing Surrogate Agent Statistics	536
SNMP Support	536
Wildcarded PUID in P-Associated-URI	536
Wildcarded Public User Identity	537
ACLI Instructions	537
Implicit Service Route	538
ACLI Instructions and Examples	538

P-CSCF: P-Asserted Identity in Responses	539
Important Notes	539
ACLI Instructions and Examples	539
P-CSCF Redundancy with Implicit Service Route	539
Tracking UE per P-CSCF	539
Registration Invalidation	540
ACLI Instructions and Examples	540
Contact Header Contact Formatting	540
ACLI Instructions and Examples	541
SIP IMS and Privacy	541
ACLI Instructions and Examples	542
Emergency Session Handling	543
ACLI Instructions and Examples	544
Recurse on 305 Only	545
8 DTMF Transfer and Support	547
DTMF Interworking	547
DTMF Indication	547
DTMF Transfer Processing Overview	548
Capability Negotiation	548
SDP Manipulated by Transcoding Policy	549
SDP Manipulated by RFC 2833 Mode	550
RFC 2833 Payload Type Mapping	552
Translation Evaluation	552
RFC 2833 Sent by Offerer	552
DTMF Audio Tones Sent by Offerer	554
SIP INFO / H.323 UII Sent By Offerer	555
Dual Mode	557
P-Dual-Info Header	558
Identical Inband with Signaling DTMF Transfer Exception	559
Override Preferred RFC 2833	560
Override Preferred DTMF Audio	560
H.323 DTMF Transfer Limitation	560
DTMF Transfer for Spiral Calls	561
P-Dual-Info Header	561
DTMF Transfer Hardware Processing	562
DTMF Transfer Configuration	562

RFC 2833 Session Agent Configuration	562
ACLI Configuration and Instructions	562
RFC 2833 Customization	565
ACLI Instructions and Examples	567
9 NAT ALG Services	569
Introduction	569
NAT ALG	569
NAPT	569
TFTP	569
Configuring the NAPT ALG	571
Configuring the TFTP ALG	573
10 DNS ALG Services	575
Introduction	575
Overview	575
Address Translation	577
Configuring DNS ALG Service	578
Before You Configure	578
ACLI Instructions and Examples	578
DNS Transaction Timeout	581
ACLI Instructions and Examples	581
DNS Server Failover	581
DNS statistics	581
11 Session Routing and Load Balancing	583
Introduction	583
Routing Overview	583
Session Agents, Session Groups, and Local Policy	583
Realms and Routing	583
About Session Agents	584
SIP Session Agents	584
SIP Session Agent Status Based on SIP Response	585
How It Works	585
ACLI Instructions and Examples	586
Managing Session Agent Traffic	586
About Session Agent Groups	587
Throttled Call Teardown to/from Session Agent	588

Accounting Considerations	588
ACLI Instructions	589
Local Policy	589
Local Policy: DTMF-Style URI Routing	590
LP Routing with ENUM provided SA Route	591
Upgrades	591
SIP Routing	591
About the Ingress Realm	592
About the Egress Realm	592
About SIP Redirect	592
SIP Method Matching and To Header Use for Local Policies	593
Load Balancing	595
Configuring Routing	596
Configuration Prerequisite	597
Configuration Order	597
Session Agent: ACLI Instructions and Examples	597
Session Agent Group: ACLI Instructions and Examples	606
Local Policy Matching for Parent Realms	611
SIP Session Agent DNS-SRV Load Balancing	612
ACLI Instructions and Examples	613
ENUM Lookup	613
How ENUM Works	613
About the Net-Net SBC ENUM Functionality	615
Configurable Lookup Length	615
Custom ENUM Service Type Support	615
ENUM Failover and Query Distribution	615
ENUM Query Distribution	615
Failover to New enum-config	616
ENUM Server Operation States	616
Server Availability Monitoring	616
ENUM Server IP Address and Port	617
Caching ENUM Responses	617
Source URI Information in ENUM Requests	617
Operation Modes	617
ACLI Instructions and Examples	619
Configuring the Local Policy Attribute	621
CNAM Subtype Support for ENUM Queries	622
CNAM Unavailable Response	623
SIP Profile Inheritance	623

ACLI Instructions and Examples	623
Local Route Tables	624
How It Works	624
HA Provisioning	625
ACLI Instructions and Examples	625
Local Route Table Support for H.323 and IWF	626
ENUM: Large UDP Datagram Support for DNS NAPTR Responses	627
Multistage Local Policy Routing	628
Routing Stages	628
Network Applications.	628
Multistage Routing Conceptual Example	628
Multistage Routing Example 2.	629
Customizing Lookup Keys.	631
Multistage Routing Lookup Termination	631
ACLI Configuration and Examples	632
Maintenance and Troubleshooting	633
Management: SNMP	633
Routing-based RN and CIC	633
ACLI Instructions and Examples	634
12 Number Translation	635
Introduction.	635
About Number Translation	635
Number Translation Configuration Overview	636
Translation Rules	636
Session Translation.	637
Applying Session Translations.	638
Configuring Number Translation	639
ACLI Instructions and Examples	639
Other Translations	641
13 Admission Control and QoS Reporting.	643
Overview	643
About Call Admission Control	643
Bandwidth-Based Admission Control	643
Session Capacity- and Rate-based Admission Control.	646
Configuring Admission Control	646
ACLI Instructions and Examples	646
Session Agent Minimum Reserved Bandwidth	649

How It Works	.650
ACLI Instructions and Examples.	.650
Aggregate Session Constraints for SIP	.651
ACLI Instructions and Examples.	.651
SIP Per-User CAC	.653
How It Works	.654
ACLI Instructions and Examples.	.655
Monthly Minutes-based CAC	.655
ACLI Instructions and Examples.	.656
Shared CAC for SIP Forked Calls	.657
Bandwidth Sharing Scenarios	.658
ACLI Instructions and Examples.	.658
Conditional Bandwidth CAC for Media Release	.659
About Conditional Bandwidth CAC for Media Release	.659
Details and Conditions.	.660
ACLI Instructions and Examples.	.661
CAC, Policing, and Marking for non-Audio/non-Video Media	.663
Configuring CAC, Policing, and Marking for non-Audio, non-Video Media.	.663
Bandwidth CAC Fallback Based on ICMP Failure	.664
Bandwidth CAC for Aggregate Emergency Sessions.	.665
ACLI Instructions and Example.	.665
About QoS Reporting	.667
Overview	.668
Configuring QoS	.669
ACLI Instructions and Examples.	.669
Network Management Controls	.669
How It Works	.669
For IWF Calls.	.671
ACLI Instructions and Examples.	.672
Management	.674
NMC Call Gapping Enhancements	.676
How It Works	.677
ACLI Instructions and Examples.	.677
Configuring the Account Configuration and Account Servers.	.678
ACLI Instructions and Examples.	.678
 14 Static Flows	 .685
Introduction.	.685

About Static Flows	685
About Network Address Translation ALG	686
Configuring Static Flows	687
Basic Static Flow Configuration Overview	687
ACLI Instructions and Examples	687
15 Security	691
Introduction	691
Security Overview	691
Denial of Service Protection	692
Levels of DoS Protection	693
About the Process	694
Trusted Path	694
Untrusted Path	695
Static and Dynamic ACL Entry Limits	696
Dynamic Deny for HNT	696
Host and Media Path Protection Process	696
Session Director Access Control	697
Media Access Control	697
Host Path Traffic Management	697
Traffic Promotion	698
Malicious Source Blocking	698
Blocking Actions	698
Protecting Against Session Agent Overloads	698
ARP Flood Protection Enhancements	698
Dynamic Demotion of NAT Devices	699
CAM Size and Flow Apportioning	700
Entry Types	700
Dynamic Reconfiguration	701
ACLI Configurations and Examples	702
Configuring DoS Protection	702
Configuration Overview	702
Changing the Default Net-Net 9200 Behavior	702
Configuring Access Control	703
Host Access Policing	706
Configuring ARP Flood Protection	707
Access Control for a Realm	708
Configuring Overload Protection for Session Agents	710
DDoS Protection from Devices Behind a NAT	711
. Restricting the Number of Endpoints behind a NAT	711

Counting Invalid Messages from Endpoints behind a NAT	711
ACLI Instructions	712
SNMP Trap support	712
Syslog Support	713
Debugging	713
Media Policing	713
Policing Methods	713
Configuration Notes	714
Configuring Media Policing for RTP Flows	715
Configuring Media Policing for Static Flows	716
Configuring Media Policing for Fragmented Packets	717
RTP Payload Type Mapping	718
TCP Synchronize Attack Prevention	718
About SYN	718
Configuring TCP SYN Attack Prevention	719
Transport Layer Security	719
The Net-Net SBC and TLS	719
TLS Features	720
Domestic and International Versions	720
Supported Encryption	721
Signaling Support	721
DoS Protection	722
Endpoint Authentication	722
Key Usage Control	723
Configuring TLS	724
Process Overview	724
Configuring Certificates	724
TLS Version	728
Configuring a TLS Profile	729
Applying a TLS Profile	730
Reusing a TLS Connection	731
Keeping Pinholes Open at the Endpoint	731
Viewing Certificates	731
TLS Cache Statistics	735
Denial of Service for TLS	735
ACLI Instructions and Examples	736
TLS Session Caching	738
ACLI Instructions and Examples	738
Untrusted Connection Timeout for TCP and TLS	739
Caveats	739

ACLI Instructions and Examples	739
TLS Statistics and Configuration	740
TLS Hardware Support	740
IPsec Support	742
Supported Protocols	742
IPsec Implementation	743
Outbound Packet Processing	743
Inbound Packet Processing	745
HA Considerations	746
Packet Size Considerations	746
IPsec Application Example	746
ACLI Instructions and Examples	747
Key Generation	751
IPsec Statistics	751
Security Association Database Commands	751
IPsec Hardware Statistics	754
show security ipsec statistics gmac	754
16 Transcoding	757
Introduction	757
Hardware Support: the Net-Net 9200 TCUs and TCMs	757
Transcoding Configuration	759
Transcoding Processing Overview	760
Defining Transcoding Policies	760
Transcoding Policy Definition	762
Syntax	762
Answer Processing and Examples	764
Unoffered Codec Reordering	764
Non-transcoded Call	764
Transcoded Call	764
Voice Transcoding	765
RFC 2833 Transcoding	771
FAX Transcoding	776
Transrating	781
Default Media Profiles	784
Supported Codecs	784
Transcodable Codecs	784
Preferred Default Payload Type	786
Redefining Codec Packetization Time	786

mptime Support for Packet Cable	786
Redefining G711FB	787
EVRC Codec Family	787
EVRC-A Specifications	788
Default settings for EVRC-A encoding:	789
EVRC-A Configuration	789
EVRC-B Specifications	789
Default/fixed settings for EVRC-B encoding:	790
EVRC-B Configuration	790
Configuring Transcoding	791
TCU Configuration	791
Transcoding Policy Configuration	792
Media Profile Configuration	795
ReINVITE for Fax Calls	796
FAX Call Detection	796
Renegotiate Timer	796
ReINVITE Codecs	796
Call Completion	796
Glare Condition	797
ACLI Instructions	798
G.726 transcoding with H.323	798
Media Profile Configuration	798
Transcoding Policy Configuration	799
G726 OLC Format	799
Media Type Subnames	800
SDP Parameter Matching	800
Using Subnames with Transcoding Policies	800
ACLI Instructions and Examples	801
 17 External Policy Servers	 803
Diameter: CAC/RACF	803
FQDN as Address for External Policy Server	803
Diameter Policy Server High Availability	803
Diameter Server States	804
HA Cluster Refresh	804
DNS Failure	804
Policy Server Failover	804
Diameter Connection	805
Diameter Heartbeat	805
Diameter Failures	806

Application IDs and Modes	806
Asynchronous SIP-Diameter Communication.	807
Serialized Diameter Messaging.	809
Bandwidth-Based Call Admission Control.	810
ACLI Instructions and Examples.	812
Additional Diameter Features	815
Experimental Result AVP	816
How It Works: RACF Experimental Result Handling	816
Optional AVP Support	817
Diameter Destination Realm AVP	819
Legacy Destination-Realm AVP Behavior	819
Signaling Only P-CSCF.	820
Media Path Deployment Models.	820
Bandwidth Request Models	821
ACLI Instructions and Examples.	822
Multi-tiered Policy Server Support	823
Messaging Details.	823
ACLI Configuration	824
HDR Support for External Policy Servers	824
18 High Availability	825
Overview	825
Intra-System HA Architecture.	825
HA Configuration and Management	826
Feature and Interface Card Redundancy	826
Health Scores	828
Failovers	829
Card State Transitions	829
Gateway Link Failure Detection and Recovery	830
Link Failure Detection	830
Link Failure Recovery	830
ARP Polling Configuration	831
NPU Failure Detection and Recovery	832
Acronym List.	835
General Use Acronyms	835
A.	835

B.....	835
C.....	836
D.....	837
E.....	838
F.....	839
G.....	839
H.....	839
I.....	839
J.....	840
K.....	840
L.....	840
M.....	841
N.....	842
O.....	842
P.....	843
Q.....	844
R.....	844
S.....	844
T.....	845
U.....	846
V.....	846
W.....	847
X.....	847
Y.....	847
Z.....	847

Appendix A: RTC Support 849

About this Guide

Overview

The *Net-Net 9000 ACLI Configuration Guide* provides information about the following:

- Basic concepts that apply to the key features and abilities of your Net-Net System
- Information about how to load the Net-Net system software image you want to use and establish basic operating parameters
- Configure system-level functionality for the Net-Net system
- Configure all components of the Net-Net 9000

Associated Documentation

The Net-Net 9200 and Net-Net 9200 OS Release S-D7.1.0 are supported by the documentation listed in the table below.

Document Number	Document Name	Description
400-0073-20	Net-Net 9200 Hardware Guide	Describes the hardware components of the Net-Net 9200, and provides removal and replacement procedures
400-0076-71	Net-Net 9000 S-D7.1.0 ACLI Configuration Guide	Provides detailed alphabetical descriptions of the commands and configurations available on the Net-Net 9200
400-0077-71	Net-Net 9000 S-D7.1.0 MIB Reference Guide	Provides information about the Net-Net 9200's MIB, trap, and alarm support
400-0078-71	Net-Net 9000 S-D7.1.0 Accounting Reference Guide	Provides information about the Net-Net 9200's RADIUS support, including Acme Packet VSAs
400-0079-71	Net-Net 9000 S-D7.1.0 Maintenance and Troubleshooting Guide	Provides information about viewing and interpreting Net-Net 9200 statistics, as well as other maintenance information
400-0080-71	Net-Net 9000 S-D7.1.0 Release Notes	Provides an overview of the Net-Net 9200 hardware and the Net-Net 9200 OS Release S-D7.1.0; includes caveats and limitations, and known issues with workarounds

Revision History

This section contains a revision history for this document.

Date	Revision Number	Description
October 25, 2010	Revision 1.00	Initial Release
December 01, 2010	Revision 1.01	- Removes section about configuring G726-16, G726-24, or G726-40 media profiles for G726 with H.323 Transcoding section - Removes unsupported trap receiver filter-level functionality
February 5, 2013	Revision 1.02	IWF section addedslow-start-no-sdp-in-invite
May 17, 2013	Revision 2.26	Changed "activate-config" to "activate config" without the hyphen to reflect how it works in the 7.1.0 software.
February 18, 2014	Revision 2.27	Corrected the Transport-Class AVP table
December 26, 2014	Revision 2.28	Removed all sections that reference unsupported refer-to-uri-prefix option.

Date	Revision Number	Description
April 27, 2012	Rev. 2.0	• Corrects typos • Corrects default values • Corrects value ranges • Corrects trunk group terminology: tgrp (and not trgp) • Updates descriptive text for QoS constraints • Adds conceptual text for "MIME Body Manipulation" Support: "How It Works: Find and Replace All" • Corrects spelling of SIP parameter: Content-type • Corrects graphic for SIP, GRUU: "Contact Header URI Replacement" • Adds descriptive text regarding QoS Reporting: RADIUS call records and HDR call records • Revises definition for register-contact-host found in the IMS Support chapter • Adds unique-sdp-id option and parameters: media-supervision-trap, and active-arp to media-manager-config found in the Security chapter • Corrects description for register-contact-host in the IMS Support chapter • Adds Note for customer-next-hop found in the IMS Support chapter • Revises valid values and adds Note for application-protocol found in the Security chapter • Revises descriptive text for session-constraints found in the Admission Control and QoS chapter • Adds minimum possible value for sustain-rate-window parameter, found in the Admission Control and QoS chapter • Adds instructional step for the permit-on-reject parameter found in "Configuring External Bandwidth Manager" in the External Policy Servers chapter • Adds Note for local-routing-config found in "Local Route Tables," in the Session Routing and Load Balancing chapter • Adds instructional steps for the disconnect-on-timeout, ingress-realm-location, and gate-spec-mask parameters found in "Configuring the External Bandwidth Manager," in the External Policy Servers chapter • Revises definition of max-sustain-window found in the Admission Control and QoS chapter • Adds Note for gateway parameter in host-route found in the System Configuration chapter • Adds definition for reuse-connections parameter found in the SIP Signaling chapter • Corrects definition and values for next-hop parameter in the Session Routing and Load Balancing chapter • Adds conceptual text for "New Reserved Word," found in the SIP Signaling chapter, "MIME Body Manipulation Support" • Adds additional values for the comparison-type parameter in the SIP Signaling chapter, "Configuring SIP Header Manipulation Element Rules" • Added conceptual text for "SIP Registration Via Proxy" and configuration instructions for the corresponding SIP option, reg-via-proxy, in the SIP Signaling chapter

Date	Revision Number	Description
		- Revises definition for source-routing in the System Configuration chapter - Adds Note to the System Configuration and Realms and Nested Realms chapters: Only one network interface can be assigned to a single realm-config object - Removes references to send-media-sessions (a session agent option) from chapters: SIP Signaling, Session Routing and Load Balancing, and Security - Adds a cross-reference in the HMR section of the SIP Signaling chapter to "HMR for SIP-ISUP" - Corrects the to-address under the Local Policy 3 column of "Multistage Routing, Example 2," found in the Session Routing and Load Balancing chapter
May 31, 2012	Rev. 2.10	- Revises definitions for ttr-no-response in the following chapters: SIP Signaling Services and H.323 Signaling Services - Revises Management Protocol Behavior, management protocols: Telnet, FTP, SFTP in non-pass-through mode, and removes SSH in pass-through mode and SFTP in pass-through mode found in Getting Started - Adds configuration step for ttr-no-response to Aggregate Session Constraints for SIP, Configuring Session Constraints in the Admission Control and Quality of Service chapter - Revises the definition for time-to-resume in the Admission Control and Quality of Service chapter
June 29, 2012	Rev. 2.20	- Removes references to SIP Interface option: strip-route-headers, and the feature, "Route Header Removal" associated with this option from the SIP Signaling Services chapter - Adds information about MIU network interfaces in the System Configuration chapter, Physical Interfaces, Management Interfaces
December 28, 2012	Rev. 2.21	- Clarifies remote IP address in MIU monitor parameter in High Availability chapter - Corrects typos in High Availability chapter
February 15, 2013	Rev: 2.25	Adds RTC Support Appendix
May 1, 2013	Rev. 2.26	- Adds DNS Server Failover - Adds max-endpoints-per-nat and nat-invalid-message-threshold to Security chapter - Adds Spanning Tree warning
May 21, 2013	Rev. 2.27	Removed READ-WRITE from access-mode in snmp-community.
September 27, 2013	Rev 2.28	Added "CNAM Subtype Support for ENUM Queries" section in Session Routing chapter and added "DDoS Protection from Devices Behind a NAT" section to Security chapter.
March 2014	Rev 2.29	Changed "For new calls, the changes will be in effect" to "A reboot is required." in RTC Appendix

Introduction

Prior to configuring your Net-Net 9200 for service, we recommend that you review the information and procedures in this chapter.

This chapter offers information that will help you:

- Review hardware installation procedures
- Connect to your Net-Net system using a console connection, Telnet, or SSH (secure shell)
- Become familiar with the Net-Net system's boot parameters and how to change them
- Manage Net-Net 9200 software licenses
- Load and activate a Net-Net system software image

Installation and Start-Up

After you have completed the hardware installation procedures outlined in the *Net-Net 9200 Hardware Installation Guide*, you are ready to establish a connection to your Net-Net system. Then you can load the Net-Net system software image you want to use and establish basic operating parameters.

Hardware Installation Summary

Installing your Net-Net system in your rack requires the steps summarized here. This list is only an overview and is not designed to substitute for following the detailed procedures in the *Net-Net 9200 Hardware Installation Guide*.

1. Unpacking the Net-Net 9200
2. Installing the Net-Net 9200 into your rack
3. Installing power supplies
4. Installing fan modules
5. Cabling the Net-Net 9200

Make sure you complete installation procedures fully and note the safety warnings to prevent physical harm to yourself and/or damage to your Net-Net 9200.

Connecting to Your Net-Net System

You can connect to your Net-Net system either through a direct console connection, or by creating a remote telnet or SSH session. Both of these access methods provide you with the full range of configuration, monitoring, and management options.

Note: By default, Telnet and FTP connections to the management interfaces on your Net-Net system are enabled.

Local Connections

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

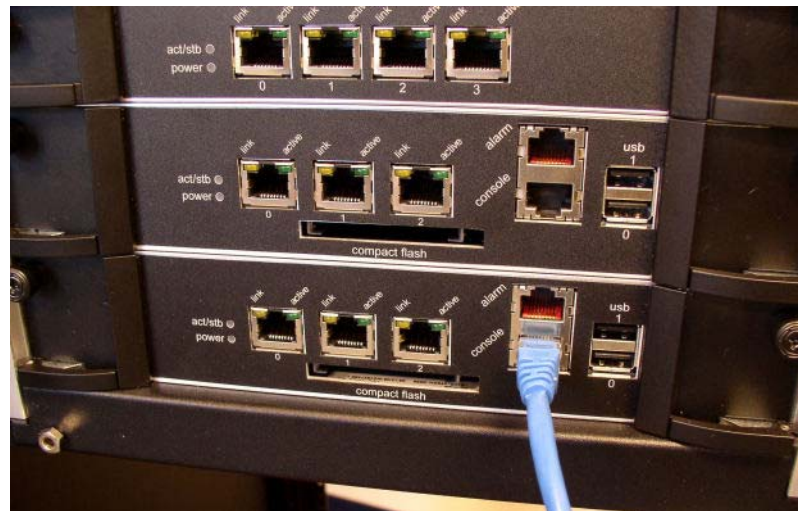
One end of the cable plugs into your terminal, and the other end plugs into the bottom RJ-45 port on right side of the MIU where the word “console” is printed in white; this provides a serial connection to the SPU opposite the MIU into which you connect. (The top RJ-45 port is used for dry contact alarms.)

To set up a console connection to your Net-Net 9200:

1. Set the connection parameters for your terminal to the default boot settings:
 - Baud rate: 115,200 bits/second
 - Data bits: 8
 - Parity: No
 - Stop bit: 1
 - Flow control: None
2. Use a serial cable to connect your PC to the Net-Net 9200.

The serial port is located on the rear of the Net-Net 9200 chassis, on right side of the MIU. It is labeled “console” in white letters, and sits directly to the left of the USB ports.

Refer to the following picture, which shows the exact location, with the serial cable inserted.



3. Power on your Net-Net 9200.
4. Enter the appropriate password information when prompted to log into User mode of the ACLI.

Telnet Remote Connections

You can also Telnet to your Net-Net 9200. Using remote Telnet access, you can provision the Net-Net 9200 remotely through the management interface over IP.

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

To Telnet to your Net-Net 9200, you need to know the IP address of its administrative interface. The IP address of your Net-Net 9200 is found by checking the **inet on ethernet** value in the boot parameters.

SSH Remote Connections

For increased security, you can connect to your Net-Net system using SSH. An SSH client is required for this type of connection.

When you connect to the Net-Net 9200 for the first time using an SSH client, you will encounter a prompt asking you to accept the Net-Net 9200's keys. Once you accept them, then your client will know about your Net-Net 9200. You will not be prompted about keys again unless the Net-Net 9200's flash memory is cleared.

The following is a sample of how certain SSH clients work; the one you select might differ.

To initiate an SSH connection to the Net-Net 9200:

1. Open your SSH client.
2. At the prompt in the SSH client, type the **ssh** command, a <Space>, the IP address of your Net-Net 9200, and then press <Enter>.

The SSH client prompts you for a password before connecting to the Net-Net 9200. Any password will be accepted.

After it is authenticated, an SSH session is initiated and you can continue with tasks in User mode or enable Superuser mode.

System Boot

When your Net-Net 9200 boots, the boot parameters tell the SPU master core what specific software image to load and where to obtain it. The SPU master core performs operations using this image; these involve decompression and extraction of the image contents.

After these operations have successfully completed, all other cores associated with the Net-Net 9200 cards use an internal control network to boot from the SPU master core. The internal network functions seamlessly so there is no need for external network access or intervention on your part.

During the booting process, the following information about the tasks and settings for the system appear:

- System boot parameters
- From which location the software image is being loaded: an external device or internal flash memory
- Hardware information including about the cards installed in chassis
- Each card starting up and registering that it is active complete
- Requisite tasks that the system is starting
- Any errors that might occur during the loading process
- Currently licensed capabilities of your Net-Net 9200

At the end of the loading process, the message **READY TO RUN!** appears.

ACLI System Prompt

The ACLI system prompt communicates information about your system's target name, level of privilege, about the configuration menu you are using, and the Net-Net 9200 core where you are working. Before you change the system target name in the boot parameters, it appears as **CLI**; the samples below show it as **ACMEPACKET**.

You can abbreviate commands that you enter at the command line to the smallest number of matching unique characters. For example, to enter the **configure terminal** section, you can simply type **conf t**.

For more information about the ACLI and how to use it, refer to the *Basics* chapter of this supplement.

- Privilege—The two levels of privilege for the Net-Net 9200 are shown in your system prompt, which is either **>** or **#**.
 - The **>** symbol means that you are in User mode, and that you have access to a certain set of monitoring and maintenance commands. When in this mode, you do not have access to configurations. You can enable Superuser mode by using the ACLI **enable** command.

The command-line prompt appears like this: **ACMEPACKET>**.

- The **#** sign means that you are in Superuser mode, and that you have access to the Net-Net 9200's full set of monitoring, maintenance, and configuration commands and settings.

You need to enter the correct password for Superuser mode in order to enable it.

The command-line prompt appears like this: **ACMEPACKET#**.

- Configuration menus—If you are in Superuser mode and have entered the **configuration terminal** section (which allows you access to the settings that to configure your system), your command-line prompt shows you where you are.

The **configure terminal** command-line prompt appears like this:
ACMEPACKET(configure)#.

Within configure terminal, you can configure different operations on your system: system, media-manager, and session-router. These prompts appear like this: **ACMEPACKET(session-router)#**.

When you start to work on different configurations on the system and enter their separate menus, the command-line system prompt reflects the specific configuration where you are working. These prompts appear like this:
ACMEPACKET(sip-nat)#.

Net-Net System Boot Parameters

Boot parameters specify the information your Net-Net 9200 uses at boot time to locate and execute its system image file. The Net-Net 9200's boot parameters:

- Configure the primary IP address for the management interface for each SPU
- Determine the software image file that a Net-Net 9200 boots from
- Specify the IP address and login credentials used for authenticating to and booting an image file from an FTP server
- Configure the virtual management IP address

In addition to providing details about the Net-Net system's boot parameters, this section explains how to view and configure them.

Note: Allow your system to boot fully before you view or start to change boot parameters; both SPUs need to have assumed their roles. Otherwise, the boot parameters you view might not be accurate; if you change any of them, then setting them might fail.

Displaying Boot Parameters

You can display the boot parameters from the ACLI using the **show bootparams** command.

To display boot parameters in the ACLI:

1. At the command-line prompt (either in User or Superuser mode), type **show bootparams** and press <Enter>.

```
ACMEPACKET# show bootparams
```

```
Boot parameters stored on ACTIVE SPU :
```

```
--- Slot 0 Boot Parameters ---
```

```
boot device      : eth
unit number     : 2
processor number : 0
host name       : nethost
file name       : nnSD700m1.tar
inet on ethernet (e) : 172.30.59.11
host inet (h)   : 172.30.0.125
gateway inet (g) : 172.30.0.1
user (u)        : xxxxx
ftp password (pw) : xxxxx
flags (f)       : 0x0
target name (tn) : ACMEPACKET
```

```
--- Slot 1 Boot Parameters ---
```

```
boot device      : eth
unit number     : 2
processor number : 0
```

```

host name           : nethost
file name           : nnSD700m1.tar
inet on ethernet (e) : 172.30.59.12
host inet (h)       : 172.30.0.125
user (u)            : xxxxx
ftp password (pw)   : xxxxx
flags (f)           : 0x0
target name (tn)    : ACMEPACKET

```

--- Virtual Management Interface Parameters ---

```

Interface IP Address: 172.30.59.13
Gateway IP Address:  172.30.0.1

```

ACMEPACKET#

The following table defines each of the Net-Net system's boot parameters.

Boot Parameter	Description
boot device	SPU1 boots from the eth2 device. SPU2 boots from the eth0 device. This value is not configurable by the user from the ACLI.
processor number	Processor number on the backplane; this should always be 0, which corresponds to the master core. This value is not configurable by the user from the ACLI.
host name	Name of the boot host used when booting from an external device.
file name	Name of the image file to be booted; can be entered with the file name path. If you are booting from the flash memory, this file name must always match the file name that you designate when you FTP the image from the source to the Net-Net 9200. When booting from flash memory, this file name must start with: /code; for example, /code/images/nnD500.tar.
inet on ethernet (e)	IP (external) address of the Net-Net 9200. In this parameter, the subnet mask ffff0000 = 255.255.0.0. You set this IP address and netmask combination in the following format: 10.21.3.5:ffff0000.
inet on backplane (b)	Internet address of the backplane interface. Leave this parameter blank when you are configuring Net-Net 9200 boot parameters.
host inet (h)	Internet address of the boot host (such as an FTP server) used when booting from an external device.
gateway inet (g)	IP address of the gateway to the boot host; this is gateway overall, and is not limited to FTP. Leave this parameter blank if the host is on the same network.
user (u)	FTP username on the boot host.
ftp password (pw)	FTP password for the FTP user on the boot host.

Boot Parameter	Description
flags (f)	Codes that signal the Net-Net 9200 from where to boot. Also signals the Net-Net system about which file to use in the booting process. This sequence always starts with 0x (these flags are hexadecimal). Enter the value 0x0 for this parameter to indicate that your system is booting from a .tar file (default).
target name (tn)	This name is required to be unique among Net-Net systems in your network, and it is reflected in the command line (name appears before > or #).
startup script (s)	For Acme Packet use only.
other (o)	For Acme Packet use only.

Changing Boot Parameters

The Net-Net 9200 supports two ways to access and change the boot parameters. You can editing them in the ACLI and then reboot the system, or you can interrupt a boot in progress before the system image has been fully executed. This section describes both methods.

Using the ACLI

You can access and change the boot parameters in ACLI Superuser mode using the **set bootparams** command and then rebooting your system. Once you have scrolled through and made any necessary changes, the system informs you that your changes will not be implemented until you reboot the system.

1. In Superuser mode, type **set bootparams** at the ACLI and press <Enter>. The first two fields, "boot device" and "file name" are automatically skipped. You can then configure the first editable boot parameter, host name, and continue from there. For example:

```
ACMEPACKET# set bootparams
```

```
Please specify boot parameters for Slot 0 :
```

```
'.' = clear field; '-' = go to previous field; ^D = quit
```

```
boot device      : eth2
processor number : 0
```

To change a boot parameter, type the new value next to the old value. To move to the next boot parameter, press <Enter>.

For example, if you want to change the host name of the system, type the new name next to the old one.

```
host name      : nethost netnetsbc
file name      : nnSD700m1.tar
inet on ethernet (e) : 172.30.59.11
inet on backplane (b):
host inet (h)   : 172.30.0.125
gateway inet (g) : 172.30.0.1
user (u)        : xxxxx
ftp password (pw) (blank = use rsh): xxxxx
flags (f)       : 0x0
target name (tn) : ACMEPACKET
```

```
startup script (s)  :
other (o)           :
```

The first time these parameters appear, they configure SPU0. After pressing <Enter> after the “other (o)” boot parameter displays, you are presented with a repeat of the same list of boot parameters that configure SPU1. All pertinent values are copied from the first time you accepted or changed the boot parameters.

Therefore, press return 12 times to reach the end of SPU1’s boot parameter list, unless you are provisioning SPU1’s network address information.

Finally you can configure the virtual management IP address. See the [Virtual Management IP Address \(46\)](#) section of this chapter for more information.

After the virtual management IP address and gateway have been changed or confirmed, the system issues the following message before returning you to the ACLI:

```
Boot parameters have been updated and will take effect on the next
reboot.
```

- For your changes to take effect, reboot the Net-Net 9200 by using the ACLI **reboot** command. Type **reboot** and then press <Enter>.

```
ACMEPACKET# reboot
```

- The Net-Net 9200 prompts you to confirm a reboot. Type a **y** and press <Enter> to continue the reboot the system. If you do not want to reboot, type an **n** and press <Enter>.

```
Are You Sure [y/n]?: y
```

The Net-Net 9200 reboots with the new boot parameters.

Interrupting a Boot in Progress

You can interrupt and halt a boot in progress on the Net-Net 9200 to change the boot parameters. This task can only be preformed from a serial connection.

To change boot parameters during a boot in progress:

- When the Net-Net 9200 boots, quickly press the Space bar on your keyboard to interrupt when you see the following message appear:

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

When the boot process halts, the [Acme Packet Boot]: prompt appears.

- At the [Acme Packet Boot]: prompt, type a **c** to change the existing boot parameters.

```
[Acme Packet Boot]: c
```

```
'.' = clear field; '-' = go to previous field; ^D = quit
```

```
boot device           : eth2
```

To navigate the boot parameters, press <Enter> and the next parameter appears on the following line. You can scroll forward, through the full list of boot parameters this way.

Caution

An *SPU1* must have its boot device configured as *eth2*; An *SPU2* must have its boot device configured as *eth0*. Failure to adhere to these guidelines will cause the Net-Net 9200 to remain unbootable.

- To go back to a previous line, type a hyphen (-) and then press <Enter>.
- Any value that you enter entirely overwrites the existing value and does not append to it.

For example, if you want to change the image you are using, type the new file name next to the old one.

```
boot device      : eth2
processor number : 0
host name       : acmepacket8
file name       : nnD500.tar nnSD700m1.tar
```

3. Continue stepping through the remaining boot parameters, changing them as needed. Then, reboot your system by using the at-sign (@) at the prompt.

[Acme Packet Boot]: @

The Net-Net 9200 immediately reboots with the new boot parameters.

Your Net-Net System Image

Your Net-Net 9200 arrives with the most recent, manufacturing-approved run-time image installed on the flash memory. If you want to use this image, you can install your Net-Net system as specified in the *Net-Net 9200 Hardware Installation Guide*, establish a connection to it, and then begin to configure it.

If you want to use an image other than the one installed on your Net-Net 9200 when it arrives, you can use the information in this section to obtain and install it.

Obtaining a New Image

You can download software images onto the platform of your Net-Net 9200 image from the SFTP site and directory where you and/or your Acme Packet customer support representative has placed images available for use.

- Obtain an image from the FTP site and directory where you and/or your Acme Packet customer support representative has placed images available for use. This may be a special server that you use expressly for images, backups, etc.
- Obtain an image from your Acme Packet customer support representative, who will transfer it to your system. Regardless source you use to obtain the image, you need to use FTP or SFTP to copy it from its source to your Net-Net system.

Using FTP to Copy an Image on Your Net-Net System

This section describes how to use FTP to copy an image to your Net-Net 9200. Copying the image can take some time, so wait for the process to complete.

To copy an image on your Net-Net 9200 using FTP:

1. Go to the directory where the image is located.

2. Check the IP address of the Net-Net 9200's management port. (You might think of this as a management address since it is used in the management of your Net-Net system.)
3. Create the connection to your Net-Net 9200. In your terminal window, type **ftp** and the IP address of your Net-Net 9200's management port, and then press <Enter>. Once a connection has been made, a confirmation note appears followed by the FTP prompt.
4. When prompted, enter your FTP username and FTP password information. The username is always *user*, and the password is the same as the one you use for the User mode login.
5. Go to the directory where you want to put the image. The */code/images* directory is used for the on-board system flash memory. This is the same directory that you specify in the boot parameters.
6. From the FTP prompt:
 - 6a. Change the directory to */code/images*.

```
ftp> cd /code/image
```
 - 6b. Invoke binary mode.

```
ftp> binary
```

Caution

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

- 6c. At the FTP prompt, type the **put** command, a <Space>, and the name of the image file. Then press <Enter>.

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

Confirmation that the connection is opening and that transfer is taking place appears.
- 6d. Wait for the file transfer is complete, and then you can quit.

```
ftp> quit
```
7. Now you are ready to boot the Net-Net 9200 using the image you just transferred.

Edit any boot parameters that need to be changed. It is especially important to change the file name boot parameter to the file name you used during the FTP process. Otherwise, your system will not boot properly.
8. Reboot your Net-Net system.
9. Your Net-Net system runs through its loading processes and return you to the ACLI command-line prompt.

System Image File Name

The system image file name is a name you set for the image. This is also the file name the boot parameters uses when booting your system. This file name must match the file name specified in the boot parameters. When you use it in the boot parameters, it must always start with */code/images* to signify that the Net-Net system is booting from the */code* directory.

If the file name set in the boot parameters does not point to the image you transferred to the Net-Net 9200, then you could not only fail to load the appropriate

image, but you could also load an image from a different directory or one that is obsolete for your purposes. If this happens, you can stop the countdown, enter the appropriate file name, and reboot the Net-Net 9200.

Booting an Image on Your Net-Net 9200

You can either boot your Net-Net 9200 from the system's flash memory or from an external device. Both locations can store images from which the system can boot. This section describes both booting methods.

For boot parameters to take effect, you must reboot your Net-Net 9200. Since a reboot stops call processing, Acme Packet recommends performing tasks that call for a reboot during off-peak maintenance hours.

Booting from Flash Memory

Once you have installed an image, you can boot your Net-Net 9200 from its flash memory. With the exception of testing an image before you install it on the flash memory, this is the most commonly used method of booting.

To boot from your Net-Net 9200 flash memory:

1. Confirm that the boot parameters are set up correctly, and make any necessary changes.
2. Change any boot configuration parameters as needed.

It is especially important to change the **file name** boot configuration parameter correctly.

```
boot device      : eth2
unit number     : 2
processor number : 0
host name       : boothost
file name       : /code/images/nnD500.tar
inet on ethernet (e) : 124.40.1.7
host inet (h)    : 124.40.0.3
gateway inet (g) : 124.40.0.1
user (u)        : vxftp
ftp password (pw) : vxftp
flags (f)       : 0x0
target name (tn) : ACMEPACKET
startup script (s) :
```

3. Reboot your Net-Net 9200. The boot process resembles the one in the following example.

```
TFFS mounting /code...
/code mounted.
Extracting from fd 4
extract acmeOS_cpu0.gz
extracting file acmeOS_cpu0.gz, size 4536969 bytes, 8862 blocks
done extracting: acmeOS_cpu0.gz
extract acmeOS_cpu1.gz
extracting file acmeOS_cpu1.gz, size 4538349 bytes, 8864 blocks
done extracting: acmeOS_cpu1.gz
end of buffers (2), done
Decompressing CPU0 image... 16653523 bytes uncompressed.
```

```

CPU0: 4 + 14011920 + 3619152
Decompressing CPU1 image... 16653523 bytes uncompressed.
CPU1: 4 + 14011920 + 3619152
/code synced and unmounted
Starting CPU0 at 0x80020000...
[Display abbreviated; full booting displays appears on your screen.]

```

If you have made an error in configuring boot parameters, the Net-Net 9200 issues an error message. Check your boot parameters, and try to reboot using the corrected set.

4. Continue with system operations.

Booting from an External Device

Booting from an external device means that your Net-Net 9200 connects to a server to retrieve the boot image at boot time. Rather than using an image stored on your system's flash memory, it downloads the image from the external device each time it reboots.

When you are testing a new image before installing it on your Net-Net 9200, you might want to boot from an external device. Ordinarily, you would not want to boot an image on your Net-Net 9200 this way.

To boot an image from an external device:

1. Confirm that the Net-Net 9200 is cabled to the network from which you are booting. The image is loaded from the source using FTP.
2. Log into the system you want to mount.
3. On the Net-Net 9200, configure the information for the boot parameters and confirm the following:
 - **boot device**—This parameter is not user-configurable.
 - **file name**—Name of the file containing the image; the image file must exist in the home directory of the "user" on the FTP server/boot host
 - **host inet**—IP address of the FTP server/boot host
 - **gateway inet**—IP address of the gateway to use if the device from which you are booting is not on the same network as your Net-Net 9200
 - **user**—Username for the FTP account on the boot host
 - **password**—Password for the FTP account on the boot host
 - **target name**—Uniquely identifies your Net-Net 9200, and sets the name you see for your system at the beginning of the command line, e.g., **ACMEPACKET#**.
4. Reboot your Net-Net 9200.
5. You are returned to the ACLI command-line prompt, and you can continue with system operations.

Setting Up System Basics

Before configuring and deploying your Net-Net system, you might want to establish some basic attributes such as adding and removing NTP servers from your system configuration and configuring time zone information on your system.

NTP Synchronization

Setting NTP synchronizes both hardware and software clocks with the NTP server(s) you specify, and it keeps the Net-Net 9200 internally synchronized.

The Net-Net 9200 acts essentially as a cluster of NTP servers with the master core at its center. The other system cores poll the master core, which keeps itself synchronized with the NTP server(s) that you configure. This means that if the master core for some reason fails to be synchronized with the NTP server(s), the Net-Net 9200's components are still synchronized with the master core and with each other.

Adding NTP Servers

To add NTP servers for NTP synchronization:

1. Enter the ACLI's configure terminal section to access configuration tasks.

```
ACMEPACKET# configure terminal
configure#
```
2. The NTP configuration is part of the ACLI's system menu. Type **system**, and then press <Enter>.

```
configure# system
system#
```
3. To reach the NTP configuration, type **ntp** and press <Enter> to access the NTP configuration. For example:

```
system# ntp
ntp#
```
4. To add an NTP server, type **server add**, a <Space>, and the IP address of the server(s) you want to add. Then press <Enter>.

For example, this entry adds the NTP servers at the Massachusetts Institute of Technology in Cambridge, MA and at the University of Massachusetts at Boston:

```
ntp# server add [NTP server 1 IP address] [NTP server 2 IP address]
ntp# done
```
5. Save your work and activate your configuration.

Removing NTP Servers

To delete NTP servers from your configuration:

1. Enter the ACLI's configure terminal section to access configuration tasks.

```
ACMEPACKET# configure terminal
configure#
```
2. The NTP configuration is part of the ACLI's system menu. Type **system**, and then press <Enter>.

```
configure# system
system#
```
3. To reach the NTP configuration, type **ntp** and press <Enter> to access the NTP configuration. For example:

```
system# ntp
ntp#
```
4. To delete an NTP server, type **server remove**, a <Space>, and the IP address of the server(s) you want to delete. Then press <Enter>.

```
ntp# server remove [NTP server 1 IP address]
```
5. Save your work and activate your configuration.

Setting the Time Zone

On your Net-Net 9200, you can set the time zone that you want the system to use. The time zone is set using Coordinated Universal Time (UTC) as a basis. When you set the time zone, you enter the number of minutes from UTC for the zone you want to use.

Configuring Time Zones

To configure a time zone settings on your Net-Net 9200:

1. Enter the ACLI's configure terminal section to access configuration tasks.

```
ACMEPACKET# configure terminal
configure#
```
2. The NTP configuration is part of the ACLI's system menu. Type **system**, and then press <Enter>.

```
configure# system
system#
```
3. To reach the NTP configuration, type **timezone** and press <Enter> to access the NTP configuration. For example:

```
system# timezone
timezone#
```
4. **name**—Set the name of the time zone using up to ten characters.
5. **minutes-from-utc**—Set the number of minutes representing the difference between the time zone you are setting and UTC. Enter a positive number of the time zone you are setting is behind UTC, and a negative number if the time zone is ahead of UTC.

The number you enter here can be as low as -720 (12 hours behind UTC) or as high as 720 (12 hours ahead of UTC).
6. **dst-start-month**—Enter the number corresponding to the month that daylight saving time (DST) starts. The default is 0.
7. **dst-start-day**—Enter the number of the day of the month when DST starts. The default is 0.
8. **dst-start-hour**—Enter the hour of the day when DST starts. The default is 0.
9. **dst-end-month**—Enter the number corresponding to the month that daylight saving time (DST) ends. The default is 0.
10. **dst-end-day**—Enter the hour of the day when DST ends. The default is 0.
11. **dst-end-hour**—Enter the hour of the day when DST ends. The default is 0.
12. Save your work and activate your configuration.

Virtual Management IP Address

You can assign a virtual management IP address to the Net-Net 9200, allowing you to set one IP address for management of the system. You set this address in the boot parameters, and you must reboot the system in order for it to take effect.

When you configure this address, RADIUS will use it; if you do not, then RADIUS uses the boot address. Syslog also uses this address if you configure it.

--- Virtual Management Interface Parameters ---

Interface IP Address:
Gateway IP Address:

To configure a virtual IP address:

1. In Superuser mode, enter **set bootparams** at the command line and press <Enter>. Starting with a help line, the boot parameters start to appear starting with the boot device.

```
ACMEPACKET# set bootparams
```

```
'.' = clear field; '-' = go to previous field; ^D = quit
```

```
boot device          : eth2
```

2. Scroll through the list of boot parameters until you reach this section:

```
--- Virtual Management Interface Parameters ---
```

```
Interface IP Address:
```

```
Gateway IP Address:
```

3. Enter the IP address (an unused virtual MAC address) for the system to use as its virtual IP address in the **Interface IP Address:** section.
4. Reboot your system in order for change to take effect.

Software Licensing

The components of the Net-Net 9200 software are licensed by Acme Packet, Inc. for your use. In order to use these components and deploy their related services in your network, you must have a valid license. There are two types of Acme Packet licenses, and you can have one of each type of license installed on your system at one time:

- **Permanent**—This license has no expiration and covers all of the features on your Net-Net 9200. If you try to add a new permanent license, the Net-Net 9200 overwrites the existing permanent license.
- **Temporary**—This license expires at specific date and time; it is active when the current date falls within the license's start and end dates. It does not matter when the expiration date occurs; every license with any expiration date is considered temporary. If you try to add a new temporary license, the Net-Net 9200 overwrites the existing temporary license.

Licenses can be activated and deactivated in real time. They are tied to specific Net-Net 9200s (by serial number) and cannot be transferred from one Net-Net 9200 to another. Before you change your licensing, the Net-Net 9200 prompts you to verify the override of the old license.

Only one license can be active on the Net-Net 9200 at one time. This means that if you want to change your licensed Acme Packet features, you must obtain a new license that will overwrite the one previously installed.

Conditions under which you might want or need a new license include:

- Changing from a temporary to a permanent license
- Adding features to your license
- Removing features from your license
- Using new features on a trial basis with a temporary license

Supported Components

The table in this section lists the software components that are licensed for the Net-Net 9200. If you do not have a license for a given component, interfaces, or feature, its configuration parameters are not visible.

Feature Group	Description
SIP	Enables SIP signaling and SIP session capacity.
QoS	Enables measurement for QoS (jitter, packet latency, and packet loss) on the Net-Net 9200.
Routing policies	Establishes routing policies on the Net-Net 9200.
Load Balancing	Establishes distribution of traffic across gateways, application servers, softswitches, etc.
Accounting	Establishes RADIUS servers to which the Net-Net 9200 can make connections and send CDRs.
ENUM	Enables Telephone Number Mapping (ENUM from Telephone Number Mapping): a suite of protocols used to unify the telephone system with the Internet by using E.164 addresses with the Domain Name System (DNS). This features requires Routing feature group to be installed.
NES (GETS)	Enables ability for the NN9200 to identify ETS/WPS/GETS calls and to provide them with the appropriate priority processing based on the Net-Net system configuration. Allows configuration of NES-related configuration object.
H.323	Enables H.323 signaling. Net-Net 9200 respond to/forward H.323 signaling messages and set up H.323 sessions based on the Net-Net system configuration. Allows configuration of H.323-related configuration objects.
IWF	Enables SIP-H.323 interworking. Net-Net 9200 responds to/forward SIP-H.323 signaling messages and set up SIP-H.323 sessions based on the Net-Net system configuration. Allows configuration of IWF-related configuration objects. Note that it is required to have both a SIP and H323 license in addition to this IWF license in order for SIP-H.323 Interworking to work.
Lawful Intercept	Configure and use parameters to enable lawful intercept (for X1/X2/X3 LI type only).
AMR	Enables transcoding to and from AMR-WB codec.
EVRC	Enables transcoding to and from EVRC0 codec.
External Policy Servers	Enables external policy server configuration and use.
External BW Mgmt	Enables external bandwidth management type (CAC / RACF) of external policy server.

Hardware-based Licenses

The following features are licensed by the installation of the hardware which enables their functionality:

Feature Group	Description
Transcoding	Enabled with the presence of the TCU/TCM.
TLS	Enabled with the presence of high speed SSM.
IPSec	Enabled with the presence of IPSec NIU.
HA	Enabled with with the presence of HA module.

Notes on Session Capacity

The Net-Net 9200 supports protocol-specific session capacity licensing.

This differs from the way that session capacity works for the Net-Net 4250. On that platform, you license a certain number of sessions that all protocols on the system use. The protocols can use them on a first-come, first-served basis.

For the Net-Net 9200, the fact that session capacity is protocol-specific means that the sessions for a specific protocol are limited to that number, and you can license the number of protocol-specific sessions best suited to your needs.

The Net-Net 9200 checks session capacity on a per-call basis. Before it sets up a call, the Net-Net 9200 confirms that there is adequate capacity for a call of that protocol type. Then it checks the capacity of the entire system and allows the call.

Unlicensed Net-Net Systems

If you log into a Net-Net system that is not licensed, you are warned that no licenses exist and that you need to enter a valid one. Until you enter a valid license, you can configure general system parameters, but not parameters for protocols and features.

When your Net-Net system arrives, you need to obtain a key to activate the license for the software components you want to use. This original set of features is enabled with one key that you obtain from Acme Packet customer support at support@acmepacket.com.

Obtaining a License

If you choose to change the licensed functionality to your Net-Net system, you need to obtain a new license from Acme Packet.

To obtain additional licenses, contact your customer support or sales representative directly or at support@acmepacket.com. You can request the changes that you want to make for your license, and Acme Packet will issue a new license to you. Then you can overwrite the old license with your new one that presents the updated feature set.

When you obtain a license, you need to provide Acme Packet with the serial number of your Net-Net system.

Trial Licenses

Acme Packet offers trial license periods for software components, allowing you to test a features before deploying them. Temporary licenses are used for this purpose.

Trial licenses are available for the same components listed in the [Supported Components \(48\)](#) section, but they only last for preset periods. After trial licenses expire, their functionality stops. At that time, you can purchase a permanent license

for the software components on your Net-Net 9200 to continue using system functionality and features.

If you have a permanent key installed on your system and you want to enter a new temporary license to try new features, the new temporary key supersedes the permanent key.

For new permanent and temporary licensing requests, contact your Acme Packet sales or customer support representative directly or at support@acmepacket.com.

Adding Licenses

Once you have obtained a key for the license, you can add the license to your Net-Net 9200. To use the license, you also need to activate it.

To add and activate a license on your Net-Net 9200:

1. In Superuser mode, enter the system section of the configure terminal menu.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
```
2. Type **license** and press <Enter> to access the licensing configuration.

```
system# license
```
3. Using the **add** command and the key generated by Acme Packet, activate the license on your Net-Net 9200. Type **add**, a <Space>, and the complete key. Then press <Enter>.

```
ACMEPACKET(license)# add s125o39pvtqhas4v2r2jc1oae9e01o21b1dmh3
```

If you enter the key incorrectly, the system returns the following error message:
failed, bad serial number (405).

4. Confirm that the license has been activated by using the ACLI **show** command within the license configuration.

The example below shows how you can have one license of each type (permanent and temporary) installed on your system.

The first license is permanent; note that it does not have an expiration date. The second license is temporary, and the example shows that it has already expired. If you wanted to resume using the functionality covered by the temporary license on a trial/temporary basis, you would need to contact Acme Packet customer support and obtain a new license. You have the option of obtaining a new temporary license for the feature, or adding the functionality to your permanently licensed feature set.

```
license# show
```

```
License #1: ad8739ef833ea48812f949ea872338dfe0923ac
32000 sessions, SIP(32000), QOS, Routing, Load Balancing, Accounting
no expiration
installed at 12:34:42 MAR 01 2006
License #2: fe3589ef833ea4884923749ea872338dfe8733ac
1000 sessions, SIP(1000) ACP
expired at 23:59:59 MAY 08 2006
installed at 12:35:43 MAR 01 2006
```

5. To activate your license, type the **activate config** command and press <Enter>. The Net-Net system then enables any of the processes that support associated features.

```
ACMEPACKET# activate config
```

Deleting Licenses

You can delete both types of licenses from your Net-Net system, including licenses that have not expired. If you want to delete a license that has not expired, you need to confirm the deletion.

For example, you might not want to choose to delete a license that has expired and that you have no plans to renew. The sample below shows how to delete a license, using an expired temporary license as an example.

To delete a license from the Net-Net system:

1. In Superuser mode, enter the system section of the configure terminal menu.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
```
2. Type **license** and press <Enter> to access the licensing configuration.

```
ACMEPACKET(system)# license
ACMEPACKET(license)#
```
3. Type the **no** command and press <Enter>. A list of possible license to delete appears.

```
ACMEPACKET(license)# no fe3589ef833ea4884923749ea872338dfe8733ac
```
4. If the license has not expired, you are be asked to confirm the deletion.

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

When you show the licenses, the one you deleted should no longer appear on the list.
5. To clear the license, type the **activate config** command and press <Enter>. The Net-Net system then disables any of the processes that support associated features.

```
ACMEPACKET# activate config
```

Temporary License Expiration

When a temporary license expires, you are no longer able to use the features associated with it. The Net-Net 9200 automatically disables all associated processes.

To avoid a temporary license unexpectedly expiring and therefore potentially disrupting service, we recommend that you track expiration dates and renew licenses well in advance of expiration. You also have the option of adding this functionality to the feature set that you license permanently, thereby avoiding the risk or expiration altogether.

Expired temporary licenses appear in your Net-Net 9200 ACLI display until you delete them, though you cannot use the features associated with them.

Viewing Licenses

There are two ways to view licenses in the ACLI.

- You can use the **show features** command at the main ACLI user prompt to see the features that are currently licensed on your Net-Net 9200.

```
ACMEPACKET# show features
32000 SIP sessions
SIP, QOS, Routing, Load Balancing, Accounting
ACMEPACKET#
```

- Within the license menu, use the **show** command to see all licenses with detailed information, including: license keys, enabled features, expiration status, expiration date (if any), and installation date.

license# **show**

License #1: ad8739ef833ea48812f949ea872338dfe0923ac

32000 sessions, SIP(32000), QoS, Routing, Load Balancing, Accounting
no expiration

installed at 12:34:42 MAR 01 2006

License #2: fe3589ef833ea4884923749ea872338dfe8733ac

1000 sessions, SIP(1000)

expired at 23:59:59 MAY 08 2006

installed at 12:35:43 MAR 01 2006

RADIUS Authentication

A security feature that extends beyond the designation of ACLI User and Superuser privileges, the User Authentication and Access control feature supports authentication using your RADIUS server(s). In addition, you can set two levels of privilege, one for all privileges and more limited set that is read-only.

User authentication configuration also allows you to use local authentication, localizing security to the Net-Net 9200 ACLI log-in modes. These modes are User and Superuser, each requiring a separate password.

The components involved in the RADIUS-based user authentication architecture are the Net-Net 9200 and your RADIUS server(s). In these roles:

- The Net-Net 9200 restricts access and requires authentication via the RADIUS server; the Net-Net 9200 communicates with the RADIUS server using either port 1812 or 1645, but does not know if the RADIUS server listens on these ports
- Your RADIUS server provides an alternative method for defining Net-Net 9200 users and authenticating them via RADIUS; the RADIUS server supports the VSA called ACME_USER_CLASS, which specifies what kind of user is requesting authentication and what privileges should be granted

How It Works

When RADIUS user authentication is enabled, the Net-Net 9200 communicates with one or more configured RADIUS servers that validates the user and specifies privileges. On the Net-Net 9200, you configure:

- What type of authentication you want to use on the Net-Net 9200
- If you are using RADIUS authentication, you set the port from which you want the Net-Net 9200 to send messages
- If you are using RADIUS authentication, you also set the protocol type you want the Net-Net 9200 and RADIUS server to use for secure communication

Although most common set-ups use two RADIUS servers to support this feature, you are allowed to configure up to six. Among other settings for the server, there is a class parameter that specifies whether the Net-Net 9200 should consider a specific server as primary or secondary. As implied by these designation, the primary servers are used first for authentication, and the secondary servers are used as backups. If you configure more than one primary and one secondary server, the Net-Net 9200 will choose servers to which it sends traffic in a round-robin strategy. For example, if you specify three servers are primary, the Net-Net 9200 will round-robin to select a server until it finds an appropriate one; it will do the same for secondary servers.

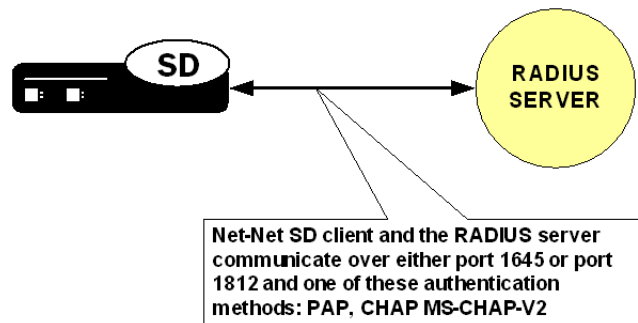
The VSA attribute assists with enforcement of access levels by containing one of the three following classes:

- None—All access denied
- User—Monitoring privileges are granted; your user prompt will resemble `ACMEPACKET>`
- Admin—All privileges are granted (monitoring, configuration, etc.); your user prompt will resemble `ACMEPACKET#`

Once it has selected a RADIUS server, the Net-Net 9200 initiates communication and proceeds with the authentication process. The authentication process between the Net-Net 9200 and the RADIUS server takes place uses one of three methods, all of which are defined by RFCs:

Protocol	RFC
PAP (Password Authentication Protocol)	B. Lloyd and W. Simpson, "PPP Authentication Protocols," RFC 1334, October 1992
CHAP (Challenge Handshake Authentication Protocol)	B. Lloyd and W. Simpson, "PPP Authentication Protocols," RFC 1334, October 1992 W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, August 1996
MS-CHAP-V2	G. Zorn, "Microsoft PPP CHAP Extensions, Version 2," RFC 2759, January 2000

Note: MS-CHAP-V2 support includes authentication only; password exchange is not supported or allowed on the Net-Net 9200.



PAP Handshake

For PAP, user credentials are sent to the RADIUS server include the user name and password attribute. The value of the "User-Password" attribute is calculated as specified in RFC 2865.

PAP Client Request Example

```
Radius Protocol
Code: Access Request (1)
Packet identifier: 0x4 (4)
Length: 61
Authenticator: 0x0000708D00002C5900002EB600003F37
Attribute value pairs
  t:User Name(1) l:11, value:"TESTUSER1"
  User-Name: TESTUSER1
```

```
t:User Password (2) l:18, value:739B3A0F25094E4B3CDA18AB69EB9E4
t:NAS IP Address(4) l:6, value:168.192.68.8
  Nas IP Address: 168.192.68.8(168.192.68.8)
t:NAS Port(5) l:6, value:118751232
```

PAP RADIUS Response

```
Radius Protocol
Code: Access Accept (2)
Packet identifier: 0x4 (4)
Length: 20
Authenticator: 0x36BD589C1577FD11E8C3B5BB223748
```

CHAP Handshake

When the authentication mode is CHAP, the user credentials sent to the RADIUS server include “username,” “CHAP-Password,” and “CHAP-Challenge.” The “CHAP-Password” credential uses MD-5 one way. This is calculated over this series of the following values, in this order: challenge-id (which for the Net-Net 9200 is always 0), followed by the user “password,” and then the challenge (as specified in RFC 1994, section 4.1).

CHAP Client Request Example

```
Radius Protocol
Code: Access Request (1)
Packet identifier: 0x5 (5)
Length: 80
Authenticator: 0x0000396C000079860000312A00006558
Attribute value pairs
  t:User Name(1) l:11, value:"TESTUSER1"
    User-Name: TESTUSER1
  t:CHAP Password (3) l:19, value:003D4B1645554E881231ED7A137DD54FBF
  t:CHAP Challenge (60) l:18, value: 000396C000079860000312A00006558
  t:NAS IP Address(4) l:6, value:168.192.68.8
    Nas IP Address: 168.192.68.8(168.192.68.8)
  t:NAS Port(5) l:6, value:118751232
```

CHAP RADIUS Response

```
Radius Protocol
Code: Access Accept (2)
Packet identifier: 0x4 (4)
Length: 20
Authenticator: 0x3BE89EED1B43D91D80EB2562E9D65392
```

MS-CHAP-v2 Handshake

When the authentication method is MS-CHAP-v2, the user credentials sent to the RADIUS server in the Access-Request packet are:

- “username”
- MS-CHAP2-Response—Specified in RFC 2548, Microsoft vendor-specific RADIUS attributes
- MS-CHAP2-Challenge—Serves as a challenge to the RADIUS server

If the RADIUS authentication is successful, the Access-Accept packet from the RADIUS server must include an MS-CHAP2-Success attribute calculated using the MS-CHAP-Challenge attribute included in the Access-Request. The calculation of MS-CHAP2-Success must be carried out as specified in RFC 2759. The Net-Net

9200 verifies that the MS-CHAP2-Success attribute matches with the calculated value. If the values do not match, the authentication is treated as a failure.

MS-CHAP-v2 Client Request Example

Some values have been abbreviated.

```
Radius Protocol
Code: Access Request (1)
Packet identifier: 0x5 (5)
Length: 80
Authenticator: 0x0000024C000046B30000339F00000B78
Attribute value pairs
  t:User Name(1) l:11, value:"TESTUSER1"
    User-Name: TESTUSER1
  t:Vendor Specific(26) l:24, vendor:Microsoft(311)
    t:MS CHAP Challenge(11) l:18, value:0000024C000046B30000339F00000B78
  t:Vendor Specific(26) l:58, vendor:Microsoft(311)
    t:MS CHAP2 Response(25) l:52, value:00000000024C000046B30000339F00000B78...
  t:NAS IP Address(4) l:6, value:168.192.68.8
    Nas IP Address: 168.192.68.8(168.192.68.8)
  t:NAS Port(5) l:6, value:118751232
```

MS-CHAP-v2 RADIUS Response

```
Radius Protocol
Code: Access Accept (2)
Packet identifier: 0x6 (6)
Length: 179
Authenticator: 0xECB4E59515AD64A2D21FC6D5F14D0CC0
Attribute value pairs
  t:Vendor Specific(26) l:51, vendor:Microsoft(311)
    t:MS CHAP Success(11) l:45, value:003533s33d3845443532443135453846313...
  t:Vendor Specific(26) l:42, vendor:Microsoft(311)
    t:MS MPPE Recv Key(17) l:36, value:96C6325D22513CED178F770093F149CBBA...
  t:Vendor Specific(26) l:42, vendor:Microsoft(311)
    t:MS MPPE Send Key(16) l:36, value:9EC9316DBFA701FF0499D36A1032678143...
  t:Vendor Specific(26) l:12, vendor:Microsoft(311)
    t:MS MPPE Encryption Policy(7) l:6, value:00000001
  t:Vendor Specific(26) l:12, vendor:Microsoft(311)
    t:MS MPPE Encryption Type(8) l:6, value:00000006
```

Management Protocol Behavior

When you use local authentication, management protocols behave the same way that they do when you are not using RADIUS servers. When you are using RADIUS servers for authentication, management protocols behave as described in this section.

- Telnet—The “user” or “admin” accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication. If authentication is successful, the user is granted privileges depending on the ACME_USER_CLASS VSA attribute.
- FTP—The “user” or “admin” accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.
- SSH—When you create an SSH account on the Net-Net 9200, you are asked to supply a user name and password. Once local authentication succeeds, you are prompted for the ACLI user name and password. The “user” or “admin” accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication. If

RADIUS authentication is successful, the privileges you are granted depend on the ACME_USER_CLASS VSA attribute.

- SFTP—The “user” or “admin” accounts are authenticated locally, not via the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.

Local Authorization and RADIUS Authentication

You can enable your Net-Net SBC to allow users administrative (Superuser) privileges, even when you have turned on RADIUS authentication. By setting the **allow-local-authorization** parameter to **enabled**, you permit users who meet the local administrative challenge criteria (i.e., enter the correct password at the prompt appearing after the ACLI **enable** command is executed) the full range of ACLI command and configuration tasks.

In situations where a RADIUS server you use for authentication cannot return the Acme Packet VSA, enabling this function gives administrators a means of, for example, making configuration changes.

Note that if a user already has administrative privileges authenticated using RADIUS, the Net-Net SBC will not challenge that user again.

ACLI Instructions and Examples

To enable RADIUS authentication and user access on your Net-Net 9200, you need to configure global parameters for the feature and then configure the RADIUS servers that you want to use.

Global Authentication Settings

To configure the global authentication settings:

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


```
ACMEPACKET# configure terminal
```
2. Type **security** and press <Enter>.


```
ACMEPACKET(configure)# security
```
3. Type **authentication** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


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

From here, you can view the entire menu for the authentication configuration by typing a ?. You can set global parameters for authentication. You can also configure individual RADIUS servers; instructions for configuring RADIUS server appear in the next section.
4. **source-port**—Set the number of the port you want to use from message sent from the Net-Net 9200 to the RADIUS server. This port can be either **1645** or **1812**; the default is 1812.
5. **type**—Set the type of user authentication you want to use on this Net-Net 9200: **local** or **radius**. The default is local.
6. **protocol**—If you are using RADIUS user authentication, set the protocol type to use with your RADIUS server(s). You have three choices: **pap**, **chap**, or **ms-chap-v2**.
7. **allow-local-authorization**—Set this parameter to **enabled** if you want the Net-Net SBC to authorize users to enter Superuser (administrative) mode locally, even when RADIUS authentication is being used.

The default for this parameter is **disabled**.

RADIUS Server Settings

The parameters you set for individual RADIUS servers identify the RADIUS server, establish a password common to the Net-Net 9200 and the server, and establish trying times.

Setting the class and the authentication methods for the RADIUS servers can determine how and when they are used in the authentication process.

To configure a RADIUS server to use for authentication:

1. Access the RADIUS server submenu from the main authentication configuration:

```
ACMEPACKET(authentication)# radius-servers
ACMEPACKET(radius-servers)#
```
2. **address**—Set the remote IP address for the RADIUS server. There is no default value, and you are required to configure this address.
3. **port**—Set the port at the remote IP address for the RADIUS server; you can use either 1645 or 1812. The default port is set to 1812.
4. **state**—Set the state of the RADIUS server. To use this RADIUS server to authenticate users, change this value from disabled (default) to enabled.
5. **secret**—Set the password that the RADIUS server and the Net-Net 9200 share. This password is transmitted between the two when the request for authentication is initiated; this ensures that the RADIUS server is communicating with the correct client.
6. **nas-id**—Set the NAS ID for the RADIUS server. There is no default for this parameter.
7. **retry-limit**—Set the number of times that you want the Net-Net 9200 to retry for authentication information from this RADIUS server. The valid range of tries is between 1 and 5, and the default value is 3.

If the RADIUS server does not respond within this number of tries, the Net-Net 9200 marks it as dead.

8. **retry-time**—Set the amount of time (in seconds) that you want the Net-Net 9200 to wait before retrying for authentication from this RADIUS server. The valid range is between 5 (default) and 10 seconds.
9. **dead-time**—Set the amount of time in seconds before the Net-Net 9200 retries a RADIUS server that it has designated as dead because that server did not respond within the maximum number of retries. The minimum is 10 seconds (default), and the maximum is 10000.
10. **maximum-sessions**—Set the maximum number of outstanding sessions for this RADIUS server. The minimum number is 1, and the maximum is 255 (default).
11. **class**—Set the class of this RADIUS server to either primary or secondary. The Net-Net 9200 tries to initiate contact with primary RADIUS servers first, and then tries the secondary servers if it cannot reach any of the primary ones.

If you configure more than one RADIUS server as primary, the Net-Net 9200 chooses the one with which it communicates using a round-robin strategy. The same strategy applies to the selection of secondary servers if there is more than one.

12. **authentication-methods**—Set the authentication method you want the Net-Net 9200 to use with this RADIUS server. Your choices are: pap (default), chap, ms-chap-v2.

This parameter has a specific relationship to the global protocol parameter for the authentication configuration, and you should exercise care when setting it. If the authentication method that you set for the RADIUS server does not match the global authentication protocol, then the RADIUS server is not used. The Net-Net 9200 simply overlooks it and does not send authentication requests to it. You can enable use of the server by changing the global authentication protocol so that it matches.

13. Save your work and activate your configuration.

Introduction

This chapter explains how to configure system-level functionality for the Net-Net 9200. Physical and network interfaces and general system parameters are required to configure your Net-Net 9200 for service.

The following configurations are explained in this chapter:

- General system parameters—Used for operating and identification purposes. In general, the informational fields have no specific effect on services, but are important to keep populated.
- Physical and network interfaces—Enable the Net-Net 9200 to communicate with other network elements. Interfaces are the most basic configurations you need to create.
- SNMP—Used for monitoring system health throughout a network.
- Syslogs and Process logs—Used to save a list of system events to a remote server for analysis and auditing purposes.
- Accounting—Establishes settings for system accounting records and allows you to add multiple RADIUS servers.

General System Information

This section explains the parameters that encompass the general system information on a Net-Net 9200. System identification information sets the system name, description, and location for your Net-Net 9200.

ACLI Instructions and Examples

To configure general system information:

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


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


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


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

System Description and Location

These parameters are optional, but it can be helpful to enter identification parameters (descriptive information) for your Net-Net 9200.

To configure the system identification:

1. **description**—Enter a description of the Net-Net 9200. This parameter is used for informational purposes.
2. **location**—Enter a location description for your Net-Net 9200. This parameter is used for informational purposes. For example, you could include the site name and address of the location where the system chassis is located.

Physical Interfaces

This section explains the different types of physical interfaces and how to configure them.

Overview

The Net-Net 9200's 10/100/1000 interfaces provide physical connections to your network. Physical interfaces are all located on the rear of the Net-Net 9200, either the MIUs or the NIUs. All cards are removable and replaceable; you can learn more about them in the *Net-Net 9200 Hardware Installation Guide*.

You need to configure operating parameters for physical interfaces to support them on your Net-Net 9200. These values identify the name, location, and basic Ethernet properties for each interface.

Types of Physical Interfaces

There are two kinds of physical interfaces on the Net-Net 9200.

- Management interfaces are on the MIU cards that populate the bottom two slots on the rear of the chassis; you can have up to two MIUs, and must have a minimum of one.

MIUs are paired with SPUs. The bottom-most MIU slot on rear of the chassis is paired with the bottom most SPU slot on the front of the chassis, and it manages that SPU. Likewise, the MIU in the next position up is paired with the next SPU. That is, MIU0 and SPU0 are paired, and MIU1 and SPU1 are paired.

You do not have to configure the MIU in the physical interface configuration; it is configured through the boot parameters.

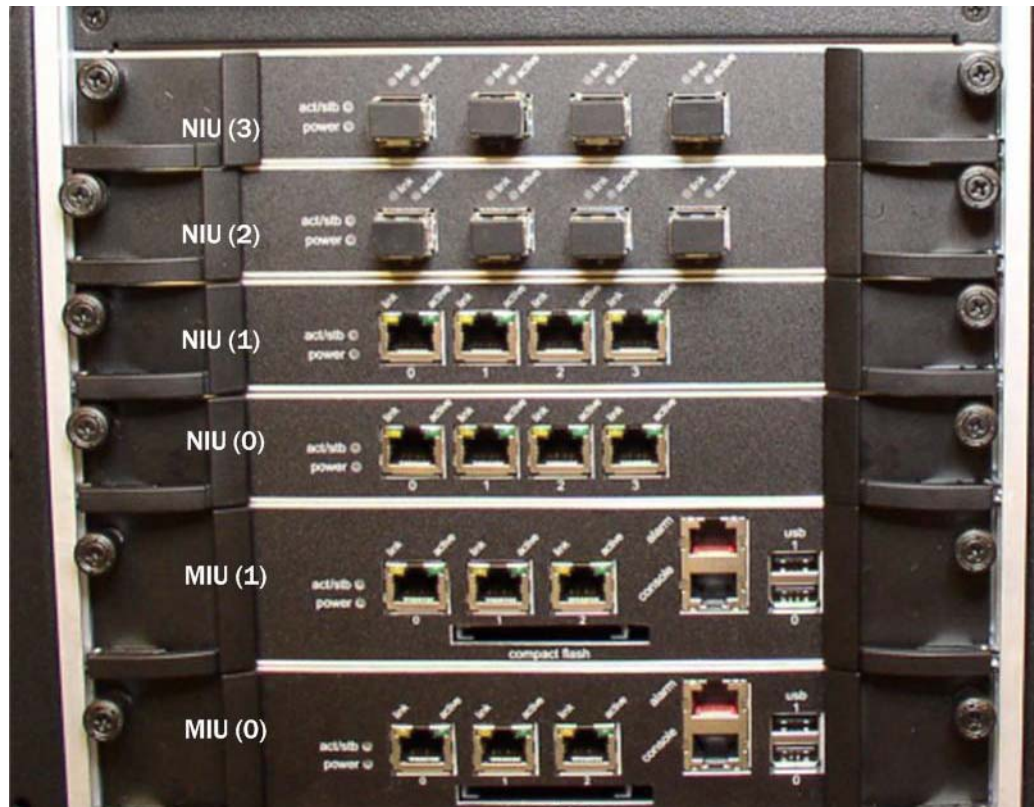
- NIUs, physical interface cards, populate the top four slots on the rear of the chassis; you can have up to four physical interface cards. The NIUs are paired for high availability; 0 and 2 begin as the active pair, while 1 and 3 begin as the standby pair. NIUs come with your choice of copper or fiber GigE, and there are some parameters that you configure only for copper.

In a four-card configuration, you only need to set the physical interface parameters for Slot 0 and for Slot 1. These cards are peers with Slot 2 and Slot 3 respectively, and so checkpoint their configuration information.

Slot and Card Location

The following picture of the rear of the Net-Net 9200 chassis shows where the MIU and NIU cards are inserted into their slots, and shows how the MIU and NIU slots are numbered.

For both MIUs and NIUs, the Net-Net 9200 slot numbering is zero-based. Numbering starts at the bottom-most card (of that card type) and increases incrementally toward the top of the system.



Although number by their slots in the photograph above, when you use the ACLI show status command, values appears they do in this table.

ACLI show status Position	Net-Net 9200 Rear Chassis Slot
7	MIU 0
8	MIU 1
9	NIU 0
10	NIU 1
11	NIU 1
12	NIU 3

NIU GigE Utilization

The Net-Net 9200's NIU physical interfaces operate in a Shared-1000Mbps per-port-pair mode. This enables each pair of physical ports on an NIU to share 1Gbps of full-duplex bandwidth. If only one of the pair's ports is used, then it is guaranteed 1 Gbps of full-duplex bandwidth. When both ports in a port pair are in-use, a single port can use more than half of the total bandwidth as long as the other port is using less than the remaining unused part of the total bandwidth (1Gbps). For example: If both ports comprising a port pair are in use, one port can burst to over 500 Mbps if its sibling pair is operating at below the 500 Mbps threshold.

The following illustration shows the matched port pairs: Port 0 and Port 2 comprise port pair A; Port 1 and Port 3 comprise port pair B.



To ensure that a port has a full 1 Gbps bandwidth available to itself, disconnect (or disable) its sibling port within the port pair.

Segmenting Traffic

In situations where a port pair should be segmented between signaling and media traffic, use the following configuration technique on the network interface port dedicated to signaling traffic:

1. Disable the **auto-negotiation** parameter.
2. Set the **speed** parameter to 100Mbps.

This technique leaves 900Mbps dedicated for media traffic.

Management Interfaces

MIU interfaces are used for management purposes such as:

- SNMP
- Telnet
- SSH
- FTP
- Logs sent from the Net-Net 9200
- Booting the Net-Net 9200 from a remote file server
- RADIUS authentication
- RADIUS accounting

MIU network interfaces are set to autonegotiate speed. The ethernet switch these interfaces connect to should also be configured with autonegotiation.

Before You Configure

Before you configure physical interfaces, decide on the number and type of physical interfaces you need. For example, you might have one media/signaling interface connecting to a private network and one connecting to the public network.

ACLI Instructions and Examples

This section describes how to configure the name, location, and Ethernet parameters for Net-Net 9200 NIU physical interfaces.

To add a physical interface:

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


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


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

```
system# phy-interface
phy-interface#
```

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

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

```
phy-interface
    name                M00
    operation-type       Media
    port                 0
    slot                 0
    admin-state          enabled
    auto-negotiation     disabled
    duplex-mode          FULL
    speed                100
ACMEPACKET(phy-interface)#
```

Identity and State

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

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

1. **name**—Set a name for the interface using any combination of characters entered without spaces. For example: **M00** (for a media and signaling interface).
2. **admin-state**—Leave the administrative state parameter set to **enabled** (default) to receive and send media and signaling on this NIU physical interface. Select **disabled** to prevent media and signaling from being received and sent on this NIU physical interface.

Operation Type and Location

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

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

1. **operation-type**—For media and signaling interfaces (which are on the NIUs), set this parameter to `Media`. This is the only valid operation type for a media/signaling interface.
2. **slot**—Set this parameter to represent the slot in which the NIU is installed.
 Entries for this parameter are 0 or 1, representing the number of NIUs you can install in the system. Slot numbering starts from the bottom NIU (slot 0) and goes upward to the top NIU (slot 3).
 Refer to the [Slot and Card Location \(60\)](#) at the top of this section for numbering and placement.
3. **port**—Set the port number.
 Like the slot numbers, port numbers are also zero-based. Numbering starts at the port furthest to the left, 0 being the left-most port and 3 being the right-most port. Refer to the photograph at the top of this section for numbering and placement.

Auto-negotiation

You need to set parameters that enable or disable auto-negotiation, set the duplex mode, and set the data rate.

Set the following parameters to configure auto-negotiation:

1. **auto-negotiation**—Leave this parameter set to `enabled` (default) so that the Net-Net 9200 and the device to which it is linked can automatically negotiate the duplex mode and speed for the link.
 If auto-negotiation is enabled, the Net-Net 9200 begins to negotiate the link to the connected device. If auto-negotiation is disabled, then the Net-Net 9200 will not engage in a negotiation of the link and will operate only at the duplex mode and speed you set.
2. **duplex-mode**—For copper NIUs only, set the duplex mode to `half` or `full`; `full` is the default.
 Given an operating speed of 100 Mbps, full duplex mode lets both devices on a link send and receive packets simultaneously using a total bandwidth of 200 Mbps. Given the same operating speed, half duplex mode limits the devices to one channel with a total bandwidth of 100 Mbps.
3. **speed**—For copper NIUs only, set the speed to `10`, `100`, or `1000` Mbps. 100 is the default.

Network Interfaces

This section describes the use and configuration of network interfaces.

Overview

The network interface element specifies a logical network interface. In order to use a network port on a network interface, you must configure both the physical interface and the corresponding network interface configuration elements. If the network interface does not use VLANs tagging, ensure that the subport ID field is set to 0, the default value. When VLAN tags are used on a network interface, the valid subport ID value can range from 1-4096. Network interfaces is a multiple instance configuration element. The combination of the name field and the subport ID field must be unique in order to identify a discrete network interface.

IP Configuration

A Net-Net 9200 network interface has standard parameters common to nearly all IP network interfaces. However, there are a few fields that are unique to the Net-Net family of products.

VLANs

VLANs are used to create a logical separation of a single physical interface into multiple network interfaces. There are several applications for this, such as:

- MPLS VPNs (RFC 2547)
- MPLS LSPs
- L2VPNs (IPsec, L2TP, ATM PVCs)
- Reusing address space
- Segmenting traffic
- Maximizing the bandwidth into a switch or router

The range of services and management capabilities you can implement by using VPNs is vast.

The primary applications of VLANs on the Net-Net 9200 are VPNs and peering. For example, several peering partners might terminate their connections to a Net-Net 9200 on a single physical interface. VLAN tags are used to segregate and correctly route the terminated traffic. The Net-Net 9200 can support a maximum of 1024 VLANs per physical interface. Ingress packets that do not contain the correct VLAN tag will be dropped. All packets exiting on an egress interface will have the VLAN tag appended to them.

The Net-Net 9200 can be included in an MPLS network through its connectivity to a PE router, which maps a MPLS VPN label to an 802.1q VLAN tag. Each Net-Net 9200, with its Gigabit Ethernet interfaces, can terminate different 802.1q VLANs into separate network interfaces. In turn, each of these separate network interfaces can represent a different customer VPN.

VLAN Network Layer Segmentation

VPNs are used to segment traffic at the network layer. Locally, a network is defined by the Net-Net 9200 as a network interface or 802.1q VLAN. Each VLAN can be bridged into a Layer 2 VPN (Ethernet VLAN, Metro VPN, ATM VC, FR DLCI), a Layer 3 Routed VPN (i.e., MPLS VPN or LSP), or may simply be used to identify a traffic class (using VLANs to segregate traffic by customer or class of service). Separation of traffic implicitly provides a level of security.

Overlapping Networks

Overlapping networks occur when two or more private networks with the same addressing schemes terminate on one physical interface.

The problem this creates can easily be solved by using VLAN tagging. For example, two 10.x.x.x networks terminating on one network interface will obviously not work properly. The Net-Net 9200 includes the IP Address, IP Subnet Mask and 802.1q VLAN tag in its network interface determination. These settings allow the Net-Net 9200 to interface directly with multiple VPNs where there is overlapping IP Address space.

Administrative Traffic

By default, the Net-Net 9200's FTP, ICMP, SNMP, and Telnet services cannot be accessed on the NIUs. In order to enable these services, the Net-Net 9200 includes

four fields that enable administrative traffic on the NIUs. These parameters are effectively firewall functions that open the well-known ports for specified services.

Network Interface Description

The network interface configuration element contains a parameter that functions as a textual description of the interface. You can enter any text in this field to label the function of the configured interface. The network interface description is limited to a maximum of 250 characters, and must be surrounded by quotation marks when the text has space characters.

Configuring Network Interfaces

This section explains how to access and configure network interface. It also provides sample configurations for your reference.

Note: Only one network-interface can be assigned to a single realm-config object.

ACLI Instructions and Examples

This section describes how to configure a network interface.

To configure a network interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **system** and press <Enter> to access the system-level configuration elements.
3. Type **network-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET# configure terminal
```

```
configure# system
```

```
system# network-interface
network-interface#
```

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

The following is an example what a network interface configuration might look like.

```
network-interface
  name M00
  sub-port-id 0
  hostname sbc.acmepacket.com
  ip-address 10.102.131.185
  netmask 255.255.255.248
  gateway 10.102.131.187
  gw-heartbeat
    state disabled
    heartbeat 1
    retry-count 0
    retry-timeout 1
    health-score 0
  dns-ip-primary 10.102.128.142
  dns-ip-backup1
  dns-ip-backup2
```

```

dns-domain                acmepacket.com
hip-ip-list                10.102.131.185
ftp                       disabled
icmp                      disabled
snmp                      disabled
telnet                    disabled
network-interface#

```

IP Configuration and Identification

You must specify the identity and address for all network interfaces.

Set the following parameters to configure a network interface:

1. **name**—Set the name for the network interface. This must be the same name as the physical interface to which it corresponds.
2. **description**—Enter a textual description for this network interface.
3. **hostname**—Set the hostname (FQDN) of this network interface. This parameter is optional.
4. **ip-address**—Set the IP address of this network interface.
5. **netmask**—Set the netmask of this network interface in dotted decimal notation.
6. **gateway**—Set the gateway that this network interface uses to communicate with the next hop.
7. **dns-ip-primary**—Set the DNS servers. You can set an additional two DNS servers by using the **dns-ip-backup1** and **dns-ip-backup2** parameters.
8. **dns-domain**—Set the default domain for use with DNS queries.

VLANs

One parameter is required to configure VLANs on a Net-Net 9200. The subport ID parameter located in the **network-interfaces** element adds and masks for a specific VLAN tag.

Set the following parameters to configure a VLAN on a network interface:

1. **sub-port-id**—Set the subport ID. If this network interface is not channelized, leave this field blank, and the value will correctly default to 0. If this network interface is channelized, enter the appropriate VLAN tag.

Enabling Management Traffic

To configure administrative service functionality, you need to enable the type of traffic you want to support on the network interface. Traffic is enabled for the IP address that you set in the **ip-address** parameter for the network interface configuration.

Set the following parameters to configure HIP functionality on a network interface:

2. **ftp**—Set this parameter to enabled to support FTP traffic, or leave it disabled (default).
3. **icmp**—Set this parameter to enabled to support ICMP traffic, or leave it disabled (default).
4. **snmp**—Set this parameter to enabled to support SNMP traffic, or leave it disabled (default).
5. **telnet**—Set this parameter to enabled to support Telnet traffic, or leave it disabled (default).

HIP IP Address List

Reserved for future use.

Secondary Gateway

A network interface on a Net-Net 9200 can be configured with a secondary default gateway. A secondary default gateway will be installed if the primary default gateway goes out of service or becomes unreachable.

Primary and secondary default gateways are monitored for connectivity by the gateway heartbeat function; the SBC's network interface sends an ARP request to a configured gateway and expects to receive an ARP reply. The secondary gateway function adds a second IP address-as-gateway to monitor and potentially use.

At boot time, the Net-Net 9200 installs whichever of the primary or secondary gateways it successfully receives an ARP reply from first. The other gateway, assuming it returns a successful ARP reply, becomes the available standby.

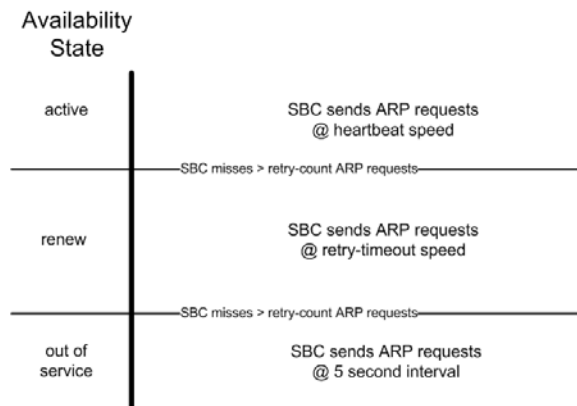
When the installed default gateway transitions to out of service state, the secondary gateway is checked for availability. If the secondary gateway is available, it is installed as the active default gateway for the network interface. In such a case, a default gateway failover on a network interface does not set an alarm.

If the secondary default gateway is out of service, and the primary default gateway is also out of service, the Gateway Unreachable alarm is triggered and the Net-Net 9200 switches to the failover NIU. Once one of the failed gateways becomes reachable, the Gateway Unreachable alarm is cleared. Gateways connected to the standby are not probed for liveliness.

Gateway Availability States

The primary and secondary default gateways transition through three states as the Net-Net 9200 monitors their connectivity:

- **active**—A gateway returns ARP replies to the Net-Net 9200's ARP requests. In active state, the Net-Net 9200 sends ARP requests to the gateway(s) at the heartbeat speed.
- **renew**—Net-Net 9200 has lost communication with a gateway by not receiving the retry-count number of consecutive ARP replies. Once in renew state, the Net-Net 9200 sends ARP requests to the gateway at the retry-timeout rate.
- **out-of-service (OOS)**—Response has not been received from the gateway after the Net-Net 9200 has sent the retry-count number of ARP requests. When a gateway exceeds the retry count in a renew state, it goes out of service. After a gateway is marked out of service, it send an ARP request to the downed gateway(s) every 5 seconds.



Configuration

The secondary gateway functionality is configured with the gateway heartbeat parameters, which apply to both primary and secondary default gateways and must be enabled for this feature to work.

- **heartbeat**—Period in seconds at which the gateways are polled for ARP connectivity when they are active and available for use.
- **retry count**—Number of missed ARP messages to transition a gateway to a renew state or an OOS state.
- **retry timeout**—Period in seconds between ARP requests when the network interface is in a renew state.
- **health score**—Health score to subtract when both gateways are unreachable and the NIU must failover.

ACLI Instructions and Examples

This section describes how to configure a secondary gateway and the timers and counters that control failover behavior.

To configure a secondary gateway on a network interface:

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


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


```
ACMEPACKET(configure)# system
```
3. Type **network-interface** and press <Enter>.


```
ACMEPACKET(system)# network-interface
```
4. Type **select** and the number of the pre-configured network interface you want to configure.


```
ACMEPACKET(system)# select 1
```

You can now configure the secondary gateway.
5. **sec-gateway**—Enter the IP address of the secondary gateway you wish to add to this network interface.
6. Save your work using the ACLI **done** command.

To configure the gateway heartbeat timers on the network interface:

Note: These values might previously have been configured. The following procedure is provided in case you wish to reconfigure heartbeat behavior.

1. Continuing from the previous step, type **gw-heartbeat** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(network-interface)# gw-heartbeat
ACMEPACKET(gw-heartbeat)#
```
2. **state**—Set this parameter to **enabled** to use the gateway heartbeat feature, or leave it **disabled** (default) to disable the feature.
3. **heartbeat**—Enter the number of seconds between heartbeats for this interface; this parameter defaults to 0. Heartbeats are sent at this interval as long as the interface is viable.

4. **retry-count**—Enter the number of heartbeat retries that you want sent to the front interface gateway before it is considered unreachable; this parameter defaults to 0.
5. **retry-timeout**—Enter the heartbeat retry time-out value in seconds; this parameter defaults to 1. This parameter sets the amount of time between Net-Net 9200 ARP requests to establish interface gateway communication after an interface gateway failure.
6. **health-score**—Enter the amount to subtract from the Net-Net 9200's health score if an interface heartbeat fails; this parameter defaults to 0. If the value you set in the retry-time-out field is exceeded, this amount will be subtracted from the system's overall health score.

SNMP

This section explains how to configure Simple Network Management Protocol (SNMP), trap receivers, and syslog servers. They provide important monitoring and system health information that contribute to a robust deployment of the Net-Net system.

Overview

SNMP is used to support monitoring of network-attached devices for conditions that warrant administrative attention. SNMP is comprised of three groups of settings on a Net-Net 9200. These settings are system-wide configurations including MIB contact information, SNMP community settings, and trap receivers.

Basic SNMP Parameters

The Net-Net 9200 includes several parameters that control basic SNMP functionality. The MIB-related elements are for informational purposes, and are helpful if set. The remainder of the parameters determines if certain Net-Net 9200 events are reported to the SNMP system.

SNMP Community

An SNMP community is a grouping of network devices and management stations used to define where information is sent and accepted. An SNMP device or agent might belong to more than one SNMP community. SNMP communities provide a type of password protection for viewing and setting management information within a community.

SNMP communities also include access level settings. They are used to define the access rights associated with a specific SNMP community. The Net-Net 9200 lets you define two types of access levels: read-only and read-write. You can define multiple SNMP communities on a Net-Net 9200 to segregate access modes per community and NMS host.

Trap Receivers

A trap receiver is an application used to receive, log, and view SNMP traps for monitoring the Net-Net 9200. An SNMP trap is the notification sent from a network device, the Net-Net 9200 in this case, that declares a change in service. Multiple trap receivers can be defined on a Net-Net 9200 either for redundancy or to segregate alarms with different severity levels to individual trap receivers.

Configuring SNMP

This section describes how to configure your Net-Net 9200 to work with external SNMP systems. Sample configurations are also provided.

SNMP Configuration Overview

1. Configure the SNMP identification information. This step includes configuring the MIB system contact, name, and location parameters.
2. Set the general SNMP parameters to enable or disable SNMP on your Net-Net 9200. Also included here are switches that govern how the SNMP system responds to specified events.
3. Set SNMP communities. Their configuration is separated into a unique configuration element.
4. Set trap receivers. Their configuration is separated into a unique configuration element.

ACLI Instructions and Examples

To configure SNMP:

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


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


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


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

From this point, you can set SNMP parameters. The following is an example what an SNMP configuration might look like. Parameters not described in this section are omitted below.

system-config

```
mib-system-contact      John Doe
mib-system-name         Test System
mib-system-location     Upstairs
snmp-enabled            enabled
enable-snmp-auth-traps  disabled
enable-snmp-syslog-notify disabled
enable-snmp-monitor-traps disabled
enable-env-monitor-traps disabled
snmp-syslog-his-table-length 1
snmp-syslog-level       WARNING
```

System Wide Configuration for SNMP

This section describes the system-wide SNMP parameters found in the System Configuration element. These parameters set global SNMP information.

Set the following parameters to configure system wide SNMP functionality:

1. **mib-system-contact**—Set the contact information used within the Net-Net system's MIB transactions. The SNMP agent sends this information to an NMS in response to an SNMP Get for the MIB-II sysContact MIB variable. This parameter's value can be a textual identification of your company's contact person for the Net-Net system and/or information about how to contact that person.
2. **mib-system-name**—Set the identification of this Net-Net 9200 presented within MIB transactions. This value, along with the target name of the Net-Net

system (identified in the boot parameters) are the values reported for MIB-II when an SNMP GET is issued by the NMS for the MIB-II sysName variable. This parameter has no direct relation to the hostname parameter in the system configuration element.

By convention, this is the node's FQDN. For SNMP MIB-II sysName GETs, the Net-Net system returns SNMP communications in the following format:

```
<targetName> [. <mib-system-name>]
```

targetName is the value configured in the target name (tn) boot parameter and mib-system-name is the value configured in this field.

3. **mib-system-location**—Set the physical location of this Net-Net 9200 that is reported within MIB transactions. This parameter is reported when an SNMP GET is issued by the NMS for the MIB-II sysLocation variable. This parameter has no direct relation to the location field in the system configuration element.
4. **snmp-enabled**—Set the SNMP system on this Net-Net 9200 to **enabled** or **disabled**. By default, this parameter is set to **enabled**.
5. **enable-snmp-monitor-traps**—When this parameter is enabled, the Net-Net 9200 generates traps with unique trap-IDs for each syslog event. If this parameter is set to disabled, a single trap-ID is used for all events, with different values in the description string. The default is disabled.
6. **enable-snmp-auth-traps**—Set whether the SNMP authentication traps are enabled. If an SNMP request fails authentication because of an IP address and SNMP community mismatch, the SNMP request will be rejected. This field determines if an SNMP trap will be sent in response to the authentication failure. Valid values for this parameter are **enabled** and **disabled**. The default is disabled.
7. **enable-env-monitor-traps**—Set whether or not the SNMP environment monitor traps are enabled. Environment traps include main board PROM temperature, CPU voltage, power supplies, fan speeds, etc.
8. **snmp-syslog-level**—Set the log severity level threshold that, if SNMP is enabled, will cause the entry to be written to the SNMP syslog history table and potentially send a syslog trap to an NMS (if **enable-snmp-auth-traps** is enabled).

SNMP Community Configuration

To configure SNMP communities:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **system** and press <Enter> to access the system-level configuration elements.
3. Type **snmp-community** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# snmp-community
ACMEPACKET(snmp-community)#
```

From this point, you can set SNMP community parameters.

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

```
snmp-community
    community-name      public
    access-mode         READ-ONLY
```


ip-addresses

10.0.1.42

Set the following parameters to configure SNMP communities:

1. **community-name**—Set the SNMP community name of an active community where this Net-Net 9200 can send or receive SNMP information. A community name value can also be used as a password to provide authentication, thereby limiting the NMSs that have access to this Net-Net system. With this field, the SNMP agent provides trivial authentication based on the community name that is exchanged in plain text SNMP messages.
2. **access-mode**—Set the access level for all NMSs defined within this SNMP community. The access level determines the permissions that other NMS hosts can wield over this Net-Net 9200. This parameter has only one option:
 - **read-only**—allows GET requests.
3. **ip-addresses**—Set one or multiple IPv4 addresses that are valid within this SNMP community. These IPv4 addresses correspond with the IPv4 address of NMS applications that monitor or configure this Net-Net 9200.

Trap Receiver Configuration**To configure trap receivers:**

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


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


```
ACMEPACKET(configure)# system
```
3. Type **trap-receiver** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# trap-receiver
ACMEPACKET(trap-receiver)#
```

From this point, you can set trap receivers.

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

```
trap-receiver
  ip-address          10.0.1.42:162
  community-name      public
```

Set the following parameters to configure trap receivers:

1. **ip-address**—Set the IPv4 address of an authorized NMS. This parameter is the IPv4 address of an NMS where traps are sent. If you do not specify a port number, the default SNMP trap port of 162 will be used.
2. **community-name**—Set the community name to which this trap receiver belongs. This community must be defined in the SNMP community element.

Syslog and Process Logs

Logging events are a critical part of diagnosing misconfigurations and optimizing operations. Net-Net 9200s can send both syslog and process log data to appropriate hosts for storage and analysis.

The Net-Net 9200 generates two types of log messages: syslog messages and process log messages. Syslog messages conform to the standard used for logging servers and processes as defined in RFC 3164. The contents of the AcmeLog (described below) essentially contains the syslog output.

Process logs are Acme Packet proprietary logs. Process logs are generated on a per-task basis and are used mainly for debugging purposes. Because process logs are more data inclusive than syslogs, their contents usually encompass syslog log data. A special application must be run on a remote server to receive process logs.

Overview of Log Files

Syslog and process log servers are both identified by an IP address and port pair.

The Net-Net OS creates and maintains two types of log files:

- The AcmeLog—This file is used to aggregate general system events that give an overview of the health and overall operating conditions of the Net-Net 9200.
- The Process Log—The process log is a file stored on a remote Acme Process Log server, on a local (to the task) core, or both. Process logs are a compendium of task log files, which are created by every discrete task that runs on the Net-Net 9200. The process log records as much or as little information as you require. The same data that shows up in a process log is also centrally available on the Net-Net 9200, accessible via FTP.

Process Log Messages

Process log messages are sent as UDP packets in the following format:

`<file-name>:<log-message>`

In this format, `<filename>` indicates the log filename and `<log-message>` indicates the full text of the log message as it would appear if it were written to the normal log file.

ACLI Instructions and Examples

This section describes how to add and remove syslog servers. You can add multiple syslog servers to your configuration.

Global Log Level

To set the global log level for the Net-Net 9200:

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


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


```
configure# system
```
3. Type **system-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
system# system-config
system-config#
```

4. **system-log-level**—Enter the global log level for the system; acmelog and syslog receive any messages that are at this level or higher. Valid values are: **ZERO**, **EMERGENCY**, **CRITICAL**, **MAJOR**, **MINOR**, **WARNING**, **NOTICE**, **INFO**. The default is **WARNING**.

Adding a Syslog Servers

To add a syslog server:

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


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


```
configure# system
```
3. Type **system-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
system# system-config
system-config#
```
4. Type **syslog-servers** and a <Space>, then **add** and a <Space>. Then enter the parameters that identify where you want to send logs: IP address, port number, and facility number.

The facility part of your entry is an integer that identifies a user-defined facility value sent in every syslog message from the Net-Net 9200 to the syslog server. This parameter is used only for identifying the source of this syslog message as coming from the Net-Net 9200. It is not identifying an OS daemon or process. The default value for this parameter is 20. RFC 3164 specifies valid facility values.

Make your entry in one command-line using the syntax in the following example:

```
ACMEPACKET(system-config)# syslog-servers add 10.168.192.5:514:17
```

When your entry is saved and you view your configuration, it looks like the following example:

```
syslog-servers          10.168.192.5:514 facility 17
```

Removing a Syslog Server

To remove a syslog server:

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


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


```
configure# system
```
3. Type **system-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
system# system-config
system-config#
```
4. Type **syslog-server** and a <Space>, then **delete** and a <Space>. Then enter the parameters that identify the location to which you want to stop sending logs: IP address, port number, and facility number. Make your entry in one command-line using the syntax in the following example:


```
system-config# syslog-servers delete 10.168.192.5:514:17
```

The syslog server is removed from your list of syslog servers.

Configuring a Process Log Server

Set the following parameters to configure the process log server:

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


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


```
configure# system
```
3. **process-log-level**—Set the starting log level all processes running on the Net-Net 9200 use. Each process running on the system has its own process log.

The default value for this parameter is **NOTICE**. Valid values for this parameter are **EMERGENCY**, **CRITICAL**, **MAJOR**, **MINOR**, **WARNING**, **NOTICE**, **INFO**, **TRACE**, **DEBUG**.

Acme Packet recommends that you do not set this parameter to a value lower than **INFO**; the recommendation is not to use **TRACE** or **DEBUG**.
4. **process-log-address**—Set the IP address of the process log server. The default value is 0.0.0.0.
5. **process-log-port**—Set the port number to use for the process log server. The default value for this parameter is 0.
6. **process-log-mode**—Set the process log mode that you want to use from these available choices:
 - **REMOTE**—Process logs sent to the server only
 - **LOCAL**—Process logs sent to the file only
 - **BOTH**—Process logs sent to remote and local
 - **NONE**—No logs are sent

Task-Specific Logging

On the Net-Net 9200, you can set log levels for specific tasks.

task-name lookup string for task [e.g. sem@0.0.0)

level logging level

mode logging mode

log-address remote log server address and port (w.x.y.z:p)

facility-logging facility-specific logging configuration

To set a log level for a specific task:

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter>.


```
configure# system
```
3. Type **task-logging** and press <Enter>


```
ACMEPACKET(system-config)# task-logging
```
4. **task-name**—Enter the task name and location of the task you are configuring in the form: <task>@<slot>.<CPU>.<core>

5. **level**—Set the log level at which the task will write event messages. If configured, this value supersedes the global RTP startup logging level, defined by the process log-level-parameter.

Valid values for this parameter are EMERGENCY, CRITICAL, MAJOR, MINOR, WARNING, NOTICE, INFO, TRACE, DEBUG. There is no default.

Acme Packet recommends that you do not set the log level for the SEM (at 0.0.0. or at 1.0.0) at anything lower than INFO; the recommendation is not to use TRACE or DEBUG.

6. **mode**—Enter the logging mode you want to use: remote, local, both, or none.

Facility Logging for Specific Tasks

You can also configure the Net-Net 9200 to perform facility-specific logging for specified tasks. You do this from within the configuration for task-specific logging; facility-logging is a submenu with the task logging configuration.

A logtype facility is a second filtering criteria used when writing to log files. The facility is a tag that describes the context of the event. Facility messages are filtered by the task and then by the mechanism that writes data to task log files to limit the amount of information that gets written into a task's log file.

To configure facility-specific logging for specified tasks:

1. Access the task logging configuration, and then type **facility-logging** and press <Enter>.


```
ACMEPACKET(system-config)# task-logging
ACMEPACKET(task-logging)# facility-logging
```
2. **facility**—Enter the facility type message you want to be logged according to the following parameters.
3. **level**—Enter the log level.
4. **mode**—Enter the mode.
5. **log-address**—Enter the remote log server address and port; your entries need to follow the IPaddress:port format.

Log Compression

The Net-Net 9200 can be configured to compress archived logs. You can set log compression globally in the system configuration, and you can also set it on a per-task basis at the command line (as opposed to setting configuration parameters) using the ACLI **set log** command.

To set log compression in the system configuration:

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


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


```
configure# system
```
3. **log-compression**—Leave this parameter set to **enabled** (default), or change it to **disabled**.

To change log compression settings using the ACLI **set** command:

1. In Superuser mode, type the ACLI **set** command and then the following on one command line with a <Space> between each:

- **log**
- **compression**
- The task for which you want to compress logs in the form task@card.cpu.core; you can wildcard any of these values using the asterisk (*) (e.g. sip*@*).

You can use partial task names to represent the whole; for example, sip is short for sipr.

- **enabled or disabled**

Your entry should resemble the following example:

```
ACMEPACKET# set log compression sip*@* disabled
```

Host Routes

This section explains how to configure host route exceptions on the Net-Net 9200.

Overview

Host routes let you insert entries into the Net-Net SD's routing table. These routes affect traffic that originates at the Net-Net SD's host process. Host routes are used primarily for steering management traffic to the correct network.

When traffic is destined for a network that is not explicitly defined on a Net-Net 9200, the default gateway (located in the system configuration) is used. If you try to route traffic to a specific destination that is not accessible through the default gateway, you need to add a host route. Host routes can be thought of as a default gateway override.

ACLI Instructions and Examples

This section describes how to configure a host route.

To configure a host route:

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


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


```
ACMEPACKET(configure)# system
```
3. Type **host-route** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# host-route
```

```
ACMEPACKET(host-route)#
```

The following is an example what a host route configuration might look like.

```
host-routes
    address          172.16.0.0
    mask             255.255.0.0
    gateway          192.168.1.1
ACMEPACKET(host-route)#
```

Three parameters define a new entry into the routing table. For each host route or routing exception you want to add, you have to make a new entry with all three of the following parameters.

Set the following parameters to configure host routes:

1. **address**—Set the IPv4 address of the destination network that this host route points toward.
2. **mask**—Set the netmask portion of the destination network for the route you are creating. The netmask is in dotted decimal notation.
3. **gateway**—Set the gateway that traffic destined for the address defined in the first two elements should use as its first hop.

Note: The gateway entered must already be defined as a gateway for an existing network interface.

4. **description**—Enter an optional description for this configuration element. If your entry contains any spaces, you must put the entire entry in double quotation marks.

Source-based Routing

The source based routing (source routing) feature is used to route packets based on their source address, and not on the system's routing table. This feature is only used for management applications within the Net-Net SBC that utilize HIP interfaces.

ACLI Instructions and Examples

Source-based routing is enabled in the system config.

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


```
ACMEPACKET# configure terminal
```
2. Type `system` and press <Enter>.


```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```
3. Type `system-config` and press <Enter>.


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

The system prompt changes to let you know that you can begin configuring individual parameters.
4. **source-routing**—Set this parameter to **enabled** to use source-based routing.
5. Save and activate your configuration.

System Access Control Lists

You can configure a system access control list (ACL) for your Net-Net SBC that determines what traffic the Net-Net SBC allows over its management interface (MIUs). By specifying who has access to the Net-Net SBC via the management interface, you can provide DoS protection for this interface.

Using a list of IP addresses and subnets that are allowable as packet sources, you can configure what traffic the Net-Net SBC accepts and what it denies. All IP packets arriving on the management interface are subject; if it does not match your configuration for system ACL, then the Net-Net SBC drops it.

Note that all IP addresses configured in the SNMP community table are automatically permitted.

ACLI Instructions and Examples

The new configuration element, `system-access-list`, is now part of the system configuration path. Its model is similar to host routes. For each entry, you must define an IP destination address and mask; you can specify either the individual host or a unique subnet.

If you do not configure this list, then there will be no ACL/DoS protection for the Net-Net SBC's management interface.

Adding an ACL for the Management Interface

To add an ACL for the management interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter>.


```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```
3. Type **system-access-list** and press <Enter>. The prompt changes to let you know you can begin configuring individual parameters.


```
ACMEPACKET(system)# system-access-list
ACMEPACKET(system-access-list)#
```
4. **source-address**—Enter the IP address representing for the source network for which you want to allow traffic over the management interface.
5. **netmask**—Enter the netmask portion of the source network for the traffic you want to allow. The netmask is in dotted decimal notation.
6. **description**—Enter an optional description for this system ACL. If your entry contains any spaces, you must put the entire entry in double quotation marks.
7. Save and activate your configuration.

Notes on Deleting System ACLs

If you delete a system ACL from your configuration, The Net-Net SBC checks whether or not there were any active FTP or Telnet clients granted access when the entry was being removed. If such a client were active during ACL removal, the Net-net SBC would warn you about the condition and ask you to confirm the deletion. If you confirm the deletion, then the Net-Net SBC's session with the active client is lost.

The following example shows you how the warning message and confirmation appear. For this example, an ACL has been deleted, and the user is activating the configuration that reflects the change.

```
ACMEPACKET# activate config
Object deleted will cause service disruption:
system-access-list: identifier=172.30.0.24

** WARNING: Removal of this system-ACL entry will result
            in the lockout of a current FTP client

Changes could affect service, continue (y/n) y

Activate Config received, processing.
```


System TCP Keepalive Settings

You can configure the Net-Net SBC to control TCP keepalives for sockets on SIP interfaces, SIP session agents, or H.323 interfaces by setting:

- The amount of time the TCP connection is idle before the Net-Net SBC starts sending keepalive messages to the remote peer
- The number of unacknowledged keepalive packets the Net-Net SBC sends before terminating the TCP connection

If TCP keepalive fails, then the Net-Net SBC will drop the call associated with that TCP connection.

In the ALCI, a configured set of network parameters appears as follows:

```
network-parameters
    tcp-keepinit-timer          75
    tcp-keepalive-count         4
    tcp-keepalive-idle-timer    400
    tcp-keepalive-interval-timer 75
    tcp-keepalive-mode          0
```

Then you apply these on a per-signaling interface or SIP session agent basis. Enabling the **tcp-keepalive** parameter in a SIP interface, H.323 Stack, or SIP Session Agent configuration element will use the values configured in the **network-parameters** configuration element for TCP keepalive status.

ACLI Instructions and Examples

TCP keepalive setting are global, and then enabled or disabled on a per-interface basis.

To configure TCP keepalive parameters on your Net-Net SBC:

Note: If you want to use the default values for TCP keepalive, you do not need to configure the **network-parameters** configuration element; you can set the **tcp-keepalive** parameter to **enabled**, and the network parameters defaults will be used.

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter> to access the system-related configurations.


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


```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```
4. **tcp-keepinit-timer**—If a TCP connection cannot be established within some amount of time, TCP will time out the connect attempt. It can be used to set the initial timeout period for a given socket, and specifies the number of seconds to wait before the connect attempt is timed out. For passive connections, this value is inherited from the listening socket. The default is **75**. The valid range is:
 - Minimum—0
 - Maximum—999999999

5. **tcp-keepalive-count**—Enter the number of packets the Net-Net SBC sends to the remote peer before it terminates the TCP connection. The default is **8**. The valid range is:
 - Minimum—0
 - Maximum— $2^{23}-1$
6. **tcp-keepalive-idle-timer**—Enter the number of seconds of idle time before TCP keepalive messages are sent to the remote peer if the **SO-KEEPALIVE** option is set. This option is set via the **h323-stack** configuration element. The default is **7200**. The valid range is:
 - Minimum—30
 - Maximum—7200
7. **tcp-keepalive-interval-timer**—When the **SO-KEEPALIVE** option is enabled, TCP probes a connection that has been idle for some amount of time. If the remote system does not respond to a keepalive probe, TCP retransmits the probe after a set amount of time. This parameter specifies the number of seconds to wait before retransmitting a keepalive probe. The default value is **75** seconds. The valid range is:
 - Minimum—15
 - Maximum—75
8. **tcp-keepalive-mode**—Set the TCP keepalive response sequence number. The default is **0**. The valid values are:
 - 0—The sequence number is sent un-incremented
 - 1—The number is incremented
 - 2—No packets are sent

Trap Event Monitoring

Two trap tables in the `ap-smgmt.mib` record trap information for any condition on the Net-Net SBC that triggers an alarm. You can poll these two tables from network management systems, OSS applications, and the Net-Net EMS to view the fault status on one or more Net-Net SBCs.

The two trap tables support alarm synchronization, and by polling them you can obtain information about the current fault condition on the Net-Net SBC. These tables are:

- **apSysMgmtTrapTable**—You can poll this table to obtain a summary of the Net-Net SBC's current fault conditions. The table records the traps that were generated as a result of the fault condition. The table is also capable of recording multiple occurrences of the same trap type that have occurred within a second of one another. For example, two or more instances of the DOS trap occurring within a second of each other are recorded. When the `apSysMgmtTrapTable` is queried, each entry in the table contains the following:
 - **NumVariables**—The amount of details recorded for the trap.
 - **SysUptime**—The SNMP `sysUptime` when the trap was generated.
 - **TrapId**—The trap ID associated with the fault condition.
- **apSysMgmtTrapInformationTable**—You can poll this table to obtain further details about the traps recorded in the `apSysMgmtTrapTable` table. When the

apSysMgmtTrapInformationTable is queried, each entry in the table contains the following:

- Data type—The SNMP information type(s) encoded in the trap.
- Data length—The octet length of the information encoded in the trap.
- The data itself (in octets)—The information represented in octets.

Trap Tables do not record information about alarm severity.

The apSysMgmtTrapTable can hold up to 1000 entries, and you can configure the number of days these entries stay in the table for a maximum of seven days. If you set this parameter to 0 days, the feature is disabled. And if you change the setting to 0 days from a greater value, then the Net-Net SBC purges the tables.

Note that the Net-Net SBC does replicate alarm synchronization table data across HA nodes.

ACLI Instructions and Examples

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type system and press <Enter>.


```
ACMEPACKET(configure)#system
ACMEPACKET(system)#
```
3. Type system-config and press <Enter>.


```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```
4. **trap-event-lifetime**—To enable alarm synchronization and cause the Net-Net SBC to record trap information in the apSysMgmtTrapTable and the apSysMgmtTrapInformationTable, set this parameter to the number of days you want to keep the information. Leaving this parameter set to 0 (default) turns alarm synchronization off, and you can keep information in the tables for up to 7 days. 7 is the maximum value for this parameter.

Accounting

This section explains SIP accounting using the Remote Authentication Dial In User Service (RADIUS) Accounting System (RAS).

The Net-Net 9200 uses the RADIUS protocol to send accounting messages. These messages are transmitted to one of a predefined list of accounting servers using a predefined forwarding strategy. RAS provides a mechanism for temporarily storing session initiation and completion statistics, and for delivering these statistics to accounting servers located elsewhere in the network.

SIP Session Accounting

The RAS client can record SIP session activity based on configuration and a CDR. The CDR determines which messages are generated and determines the RADIUS attributes included in the messages. The RAS client must be capable of sending CDRs to any number of RADIUS accounting servers, using the defined hunt, failover, round robin, fewest pending, or fastest server strategies.

The establishment, failed establishment, change, or removal of a SIP session can trigger RADIUS Accounting Request messages. The RAS might also send

notification of its status (enabled/disabled). RADIUS Accounting Request messages include the following:

- **Start**—SIP session has started.
- **Interim-Update**—SIP session parameters have changed.
- **Stop**—SIP session has ended.
- **Accounting-On**—Creation of a new RADIUS client.
- **Accounting-Off**—RADIUS client has shut down.

Each SIP session might generate Start, Interim-Update, and Stop messages based on the local configuration when the session is initiated. Each Start message tells the RADIUS server that a SIP session has started. Each Interim-Update message changes the SIP session parameters, and may report the session characteristics for the SIP session to that point. Each Stop message informs the RADIUS server that a SIP session has ended and reports session characteristics.

The RAS can transmit all RADIUS messages related to a SIP session at the end of the session, regardless of which messages are generated and when they are generated. You might choose this option to reduce the likelihood of the RADIUS messages being logged to different servers, or in different log files on the same server.

The RAS always generates a RADIUS Stop message when the SIP session ends, regardless of the SIP session termination cause. The termination cause and the session characteristics are reported.

Your accounting configuration might resemble this sample:

```

account-config
  hostname                localhost
  port                    1813
  strategy                 Hunt
  state                    enabled
  max-msg-delay            60
  max-wait-failover        100
  trans-at-close           disabled
  generate-start           OK
  generate-interim         Reinvite-Response
  account-servers
    hostname              10.102.128.143
    port                  1813
    state                  enabled
    min-round-trip        250
    max-inactivity         60
    restart-delay          30
    bundle-vsa             enabled
    NAS-ID                 ACMEPACKET

```

ACLI Instructions and Examples

To configure the accounting configuration and accounting servers:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
configure# session-router
```

3. Type **account-config** and press <Enter>.


```
session-router# account-config
account-config#
```
4. To configure accounting server parameters (a subset of the accounting configuration), type **account-servers** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
account-config# account-servers
account-server#
```

Accounting Configuration

You set the accounting configuration parameters to indicate where you want accounting messages sent, when you want them sent, and the strategy you want used to select accounting servers.

To configure the accounting configuration:

1. **hostname**—Defaults to and must remain `localhost`.
2. **port**—Retain the default value of 1813 or enter the number of the UDP port associated with the Net-Net system from which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535
3. **strategy**—Set the strategy you want used to select the accounting servers to which the Net-Net system will send its accounting messages. The following table lists the available strategies:

Strategy	Description
hunt	Selects accounting servers in the order in which they are listed. If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers
failover	Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
round robin	Selects each accounting server in order, distributing the selection of each accounting server evenly over time.
fastest round trip time	Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
fewest pending	Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the Net-Net system).

4. **state**—Retain the default value `enabled` if you want the accounting configuration active on the system. Enter `disabled` if you do not want the accounting configuration active on the system.

5. **max-msg-delay**—Retain the default value of **60** seconds or indicate the length of time in seconds that you want the Net-Net system to continue trying to send each accounting message. During this delay, the Net-Net system can hold a generic queue of 4096 messages.

- minimum: zero (0)
- maximum: 999999999

6. **max-wait-failover**—Retain the default value of **100** messages or set the maximum number of accounting messages the Net-Net 9200 can store its message waiting queue for a specific accounting server before it is considered a failover.

Once this value is exceeded, the Net-Net 9200 attempts to send its accounting messages—including its pending messages—to the next accounting server in its configured list.

- minimum: one (1) message
- maximum: 4096 messages

7. **trans-at-close**—Retain the default value of **disabled** if you do not want to defer the transmission of message information to the close of a session. Enter **enabled** if you want to defer message transmission.

- **disabled**—Net-Net 9200 transmits accounting information at the start of a session (Start), during the session (Interim), and at the close of a session (Stop). The transmitted accounting information for a single session might span a period of hours and be spread out among different storage files.
- **enabled**—Limits the number of files on the Net-Net 9200 used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.

8. **generate-start**—Set the event that causes a RADIUS Start record to be generated.

- Set **None** so that Start records will not be generated
- Set **INVITE** so that Start records will be generated on receipt of the INVITE
- Set **OK** so that Start records will be generated on receipt of the OK

9. **generate-interim**—Set the list of events that generate the RADIUS Interim record, such as: OK, REINVITE, REINVITE-RESPONSE, REINVITE-CANCEL. You can edit this list using the **add** and **delete** commands.

10. **account-server**—Create the accounting server list to store accounting server information for the accounting configuration. Each accounting server can hold 100 accounting messages. See the next section for step-by-step instructions.

Accounting server entries are specific to the account configuration. They cannot be viewed or accessed for editing outside of the accounting configuration.

Note: RADIUS accounting will not be enabled if you do not enter one or more servers in the list.

Accounting Server

You must establish the list of servers to which the Net-Net system can send accounting messages.

1. **hostname**—Name of the host associated with the accounting server in hostname format (FQDN) or as an IP address.

2. **port**—Retain the default **1813** or enter the number of the UDP port associated with the accounting server to which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535
3. **state**—Retain the default **enabled** to enable the accounting servers on the system or enter **disabled** to disable them.
4. **min-round-trip**—Retain the default **250** milliseconds or indicate the minimum round trip time of an accounting message.
 - minimum: 1025 milliseconds
 - maximum: 65535 milliseconds

A round trip consists of the following:

- The Net-Net system sends an accounting message to the accounting server.
- The accounting server processes this message and responds back to the Net-Net system.

If the fastest RTT is the strategy for the accounting configuration, the value you enter here can be used to determine an order of preference (if all the configured accounting servers are responding in less than their minimum RTT).

5. **max-inactivity**—Retain the default **60** seconds or indicate the length of time in seconds that you want the Net-Net system with pending accounting messages to wait when it has not received a valid response from the target accounting server.
 - minimum: 1 second
 - maximum: 300 seconds

Once this timer value is exceeded, the Net-Net system marks the unresponsive accounting server as disabled in its failover scheme. When a server connection is marked as inactive, the Net-Net system attempts to restart the connection and transfers pending messages to another queue for transmission. RADIUS messages might be moved between different accounting servers as servers become inactive or disabled.

6. **restart-delay**—Retain the default **30** seconds or indicate the length of time in seconds you want the Net-Net system to wait before resending messages to a disabled accounting server.
 - minimum: 1 second
 - maximum: 300 seconds
7. **bundle-vs-a**—Retain the default **enabled** if you want the accounting server to bundle the VSAs within RADIUS accounting messages. Enter **disabled** if you do not want the VSAs to be bundled. (Bundling means including multiple VSAs within the vendor value portion of the message.)

In a bundled accounting message, the RADIUS message type is vendor-specific, the length is determined for each individual message, and the vendor portion begins with a 4-byte identifier, and includes multiple vendor type, vendor length, and vendor value attributes.

8. **secret**—Enter the secret passed from the accounting server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IP address of the received packet.

9. **NAS-ID**—*Optional*. Enter the NAS ID in text format (FQDN allowed). The accounting server uses this value to identify the Net-Net system for the transmittal of accounting messages.

The remote server to which the accounting configuration sends messages uses at least one of two potential pieces of information for purposes of identification. The Net-Net system accounting messages always includes in the first of these:

- Network Access Server (NAS) IP address (the IP address of the Net-Net system's SIP proxy)
- NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.
- NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

If you have more than one Net-Net 9200 pointing to the same accounting server, the NAS ID can be used to identify which Net-Net 9200 generated the record.

Introduction

This chapter explains how to configure realms and nested realms.

A realm is a logical definition of a network or groups of networks made up in part by devices that provide real-time communication sessions comprised of signaling messages and possibly media flows. These network devices might be call agents, softswitches, SIP proxies, IP PBXs, etc., that are statically defined by IP addresses. These network devices might also be IP endpoints: SIP phones, IADs, MAs, media gateways, etc., that are defined by an IP address prefix.

Realms support bandwidth-based call admission control and QoS marking for media. They are the basis for defining egress and ingress traffic to the Net-Net 9200—which supports the Net-Net 9200's topology hiding capabilities.

This chapter also explains how to configure media ports (steering pools). A steering pool exists within a realm and contains a range of ports that have a common address (for example, a target IP address). The range of ports contained in the steering pool are used to steer media flows from one realm, through the Net-Net 9200, to another.

Finally, in this chapter you can learn about TOS/DiffServ functionality for realm-based packet marking by media type.

Overview

Realms are a logical distinction representing routes (or groups of routes) reachable by the Net-Net 9200 and what kinds of resources and special functions apply to those routes. Realms are used as a basis for determining ingress and egress associations to network interfaces, which can reside in different VPNs. The ingress realm is determined by the signaling interface on which traffic arrives. The egress realm is determined by the following:

- Routing policy—Where the egress realm is determined in the session agent configuration or external address of a SIP-NAT
- Realm-bridging—As applied in the SIP-NAT configuration
- Third-party routing/redirect (i.e., SIP redirect)

Realms also provide configuration support for denial of service (DoS)/access control list (ACL) functionality. For more information about the Net-Net 9200's DoS/ACL capabilities and configuration, refer to this guide's *Security* chapter.

Realms can also be nested in order to form nested realm groups. Nested realms consist of separate realms that are arranged within a hierarchy to support network architectures that have separate backbone networks and VPNs for signaling and media. This chapter provides detailed information about nested realms after showing you how to configure realms on your Net-Net 9200.

About Realms and Network Interfaces

All realms reference network interfaces on the Net-Net 9200. This reference is made when you configure a list of network interfaces in the realm configuration.

You configure a network interface to specify logical network interfaces that correspond existing physical interfaces on the Net-Net 9200. Configuring multiple network interfaces on a single physical interface creates a channelized physical interface, a VLAN. VLANs, in turn, allow you to reuse address space, segment traffic, and maximize bandwidth.

In order to reach the realms you configure, you need to assign them network interfaces. The values you set for the name and port in the network interface you select then indicate where the realm can be reached.

About the SIP Home Realm

The realm configuration is also used to establish what is referred to as the SIP home realm. This is the realm where the Net-Net 9200's SIP proxy sits.

In peering configurations, the SIP home realm is the internal network of the SIP proxy. In backbone access configurations, the SIP home realm typically interfaces with the backbone connected network. In additions, the SIP home realm is usually exposed to the Internet in an HNT configuration.

Although you configure a SIP home realm in the realm configuration, it is specified as the home realm in the main SIP configuration by the home realm identifier parameter. Specifying the SIP home realm means that the Net-Net 9200's SIP proxy can be addressed directly by connected entities, but other connected network signaling receives layer 3 NAT treatment before reaching the internal SIP proxy.

For more information about SIP functionality and features, refer to this guide's *SIP Configuration* chapter.

About Realms and Other Net-Net 9200 Functions

Realms are referenced by other configurations in order to support this functionality across the protocols the Net-Net 9200 supports and to make routing decisions. Other configurations' parameters that point to realms are:

- SIP configuration: home realm identifier, egress realm identifier
- SIP-NAT configuration: realm identifier
- Session agent configuration: realm identifier
- Media manager: home realm identifier
- Steering ports: realm identifier
- Static flow: in realm identifier, out realm identifier

Configuring Realms

Realm configuration is divided into the following functional areas, and the steps for configuring each are set out in this chapter: identity and IP address prefix, realm interfaces, realm service profiles, QoS measurement, QoS marking, address translation profiles, and DNS server configuration.

Before You Configure

Before you configure realms, you want to establish the physical and network interfaces with which the realm will be associated.

- Configure a physical interface to define the physical characteristics of the signaling line.
- Configure a network interface to define the network in which this realm is participating and optionally to create VLANs.

If you wish to use QoS, you should also determine if your Net-Net 9200 is QoS enabled.

Remember that you will also use this realm in other configurations to accomplish the following:

- Set a signaling port or ports at which the Net-Net 9200 listens for signaling messages.
- Configure sessions agents to point to ingress and egress signaling devices located in this realm in order to apply constraint for admission control.
- Configure session agents for defining trusted sources for accepting signaling messages.

ACLI Instructions and Examples

To access the realm configuration parameters in the ACLI:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

Identity and IP Address Prefix

The first parameters you configure for a realm are its name (a unique identifier) and an IP address prefix and subnet mask.

The IP address and subnet mask establish a set of matching criteria for the realm, and distinguishes between realms that you assign to the same network interface.

To configure a realm's identity and IP address prefix in the ACLI:

1. **identifier**—Enter the name of the realm. This parameter uniquely identifies the realm. You will use this parameter in other configurations when asked for a realm identifier value.
2. **description**—Enter an optional description for this realm. If your entry contains any spaces, you must put the entire entry in double quotation marks.
3. **addr-prefix**—Enter the IP address and subnet mask combination to set the criteria the Net-Net 9200 uses to match packets sent or received on the network interface associated with this realm. This matching determines the realm, and subsequently what resources are used for that traffic.

This parameter must be entered in the correct format where the IP address comes first and is separated by a slash (/) from the subnet mask value. For example, 172.16.0.0/24.

If you leave this parameter set to its default, 0.0.0.0, then all addresses will match.

Realm Interfaces

The realm points to one network interface on the Net-Net 9200. For more information, refer to this chapter's *About Realms and Network Interfaces* section. For information about configuring network interfaces and VLAN support, refer to this guide's *System Configuration* chapter.

Note: Only one network-interface can be assigned to a single realm-config object.

You can edit the network interfaces for a realm that already exists by using the **add** and **delete** commands.

To assign interfaces to a realm:

1. **network-interfaces**—Enter the physical and network interface(s) that you want this realm to reference. These are the network interfaces through which this realm can be reached by ingress traffic, and through which this traffic exits the system as egress traffic.

Enter the name and port in the correct format where the name of the interface comes first and is separated by a colon (:) from the port number. For example, gige1:0.

The parameters you set for the network interfaces must be unique.

Enter multiple network interfaces for this list by typing an open parenthesis, entering each field value separated by a <Space>, typing a closed parenthesis, and then pressing <Enter>.

```
ACMEPACKET# network-interfaces (gige1:0 gige2:1)
```

To add network interfaces to a realm:

1. Type the following: **network-interfaces** and a <Space>, the **add** command and a <Space>, and then the network interface and port combination. Press <Enter>.

```
ACMEPACKET# network-interfaces add gige2:3
```

Enter multiple network interfaces for this list by typing an open parenthesis, entering each field value separated by a <Space>, typing a closed parenthesis, and then pressing <Enter>.

To delete network interfaces to a realm:

1. Type the following: **network-interfaces** and a <Space>, the **delete** command and a <Space>, and then the network interface and port combination. Press <Enter>.

```
ACMEPACKET(realm-config)# network-interfaces delete gige2:3
```

Enter multiple network interfaces for this list by typing an open parenthesis, entering each field value separated by a <Space>, typing a closed parenthesis, and then pressing <Enter>.

Realm Service Profile

The parameters you configure to establish the realm service profile determine how bandwidth resources are used and how media is treated in relation to the realm. Bandwidth constraints set for realm service profiles support the Net-Net 9200's admission control feature. For further information about this feature, refer to this guide's *Admission Control and QoS* chapter.

Peer-to-peer media between endpoints can be treated in one of three different ways:

- Media can be directed between sources and destinations within this realm on this specific Net-Net 9200. Media travels through the Net-Net 9200 rather than straight between the endpoints.
- Media can be directed through the Net-Net 9200 between endpoints that are in different realms, but share the same subnet.
- For SIP only, media can be released between multiple Net-Net 9200s.

To enable SIP distributed media release, you must set the appropriate parameter in the realm configuration. You must also set the SIP options parameter to media-release with the appropriate header name and header parameter information. This option defines how the Net-Net 9200 encodes IP address and port information for media streams described by, for example, SDP.

To configure realm service profile:

1. **max-bandwidth**—Enter the maximum bandwidth resource available for this realm. The default is 0.
2. **mm-in-realm**—Set this parameter to **enabled** (default is **disabled**) to treat media within this realm on this Net-Net 9200.
3. **mm-in-network**—Leave this parameter set to **enabled** (default) to treat media within realms that have the same subnet mask on this Net-Net 9200.
4. **msm-release**—Set this parameters to **enabled** (default is **disabled**) to support the SIP distributed media release feature. Remember that for this feature to work, you must also set the options parameter in the SIP configuration accordingly.

QoS Measurement

Refer to this guide's *Admission Control and QoS* chapter for more information about enabling QoS measurements on your Net-Net 9200. This chapter provides detailed information about when to configure the **qos-enable** parameter. If you are not using QoS or a QoS-capable Net-Net 9200, then you can leave this parameter set to **disabled** (default).

QoS Marking

QoS marking allows you to apply a set of TOS/DiffServ mechanisms that enable you to provide better service for selected networks

You can configure a realm to perform realm-based packet marking by media type, either audio/voice or video.

The realm configuration references a set of media policies that you configure in the media policy configuration. Within these policies, you can establish TOS/DiffServ values that define an individual type (or class) of service, and then apply them on a per-realm basis. In the media profiles, you can also specify:

- One or more audio media types for SIP
- One or more video types for SIP
- Both audio and video media types for SIP

To establish what media policies to use per realm in the ACLI:

1. **media-policy**—Enter the name (unique identifier) of the media policy you want to apply in the realm. If you leave this parameter empty, then QoS marking for media will not be performed for this realm.

Address Translation Profiles

Refer to this guide's *Number Translations* chapter for realm-specific information about using address translations on your Net-Net 9200. If you are not using this feature, you can leave the **in-translationid** and **out-translationid** parameters blank.

DNS Servers

You can configure DNS functionality on a per-network-interface basis, or you can configure DNS servers to use per realm. Configuring DNS servers for your realms means that you can have multiple DNS servers in connected networks. In addition, this allows you to specify which DNS server to use for a given realm such that the DNS might actually be in a different realm with a different network interface.

To configure realm-specific DNS in the ACLI:

1. **dns-realm**—Enter the name of the network interface that is configured for the DNS service you want to apply in this realm. If you do not configure this parameter, then the realm will use the DNS information configured in its associated network interface.

DoS/ACL Configuration

Refer to this guide's *Security* chapter for realm-specific information about using DoS/ACL functionality on your Net-Net 9200. If you are not using this functionality, you can leave the parameters at their default values: **average-rate-limit**, **peak-rate-limit**, **maximum-burst-size**, **access-control-trust-level**, **invalid-signal-threshold**, and **maximum-signal-threshold**.

Admission Control Configuration

You can set admission control based on bandwidth for each realm by setting the **max-bandwidth** parameter for the realm configuration. Details about admission control are covered in this guide's *Admission Control and QoS* chapter.

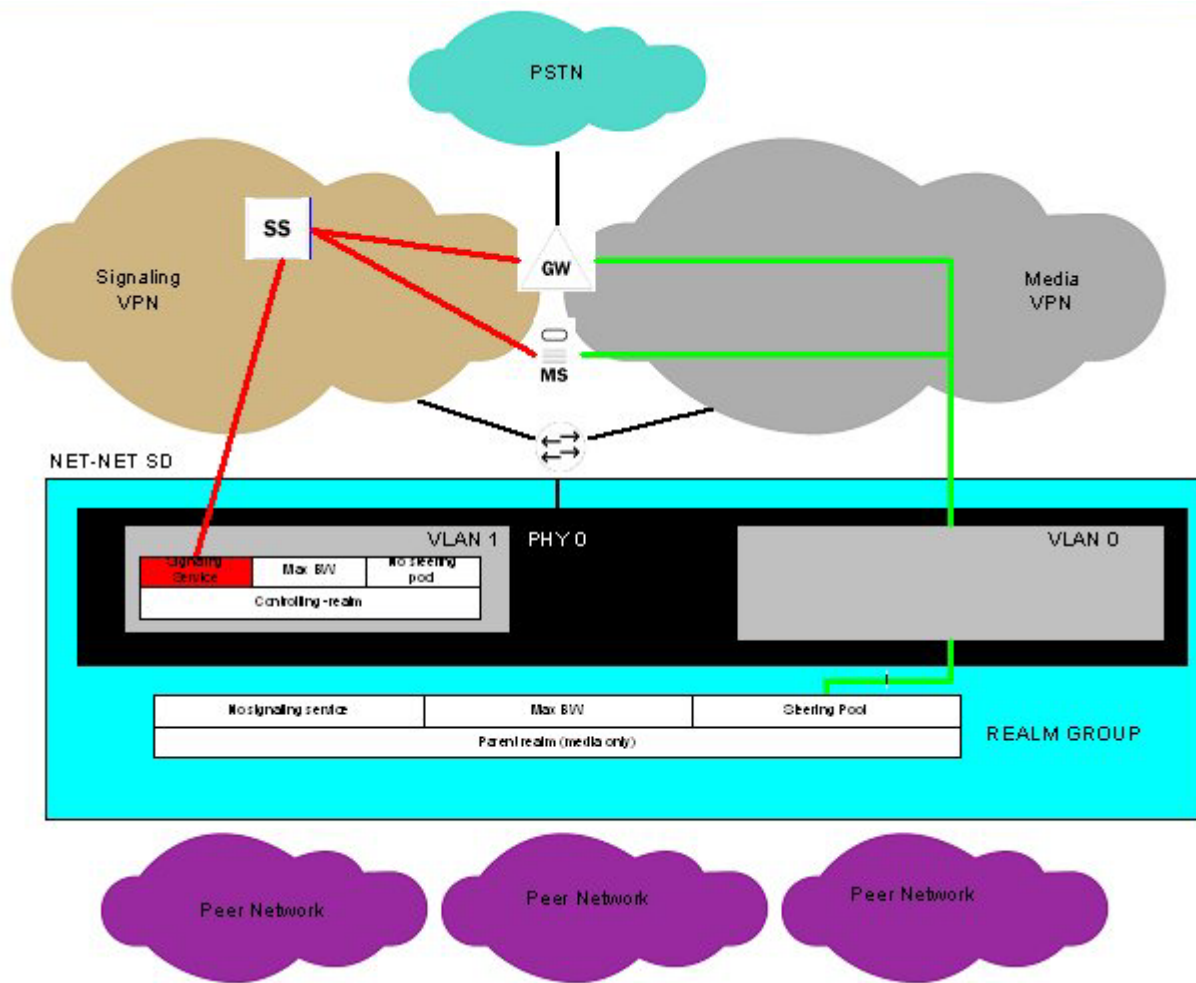
Reserved Parameters

In the ACLI, you do not need to configure the following parameters: **max-latency**, **max-jitter**, **max-packet-loss**, and **observ-window-size**.

Nested Realms

Configuring nested realms allows you to create backbone VPN separation for signaling and media. This means that you can put signaling and media on separate network interfaces, that the signaling and media VPN can have different address spaces, and that the parent realm has one media-only sub-realm.

The following figure shows the network architecture.



In addition, you can achieve enhanced scalability by using a shared service interface. A single service address is shared across many customers/peers, customer specific policies for bandwidth use and access control are preserved, and you can achieve fine-grained policy control.

These benefits are achieved when you configure these types of realms:

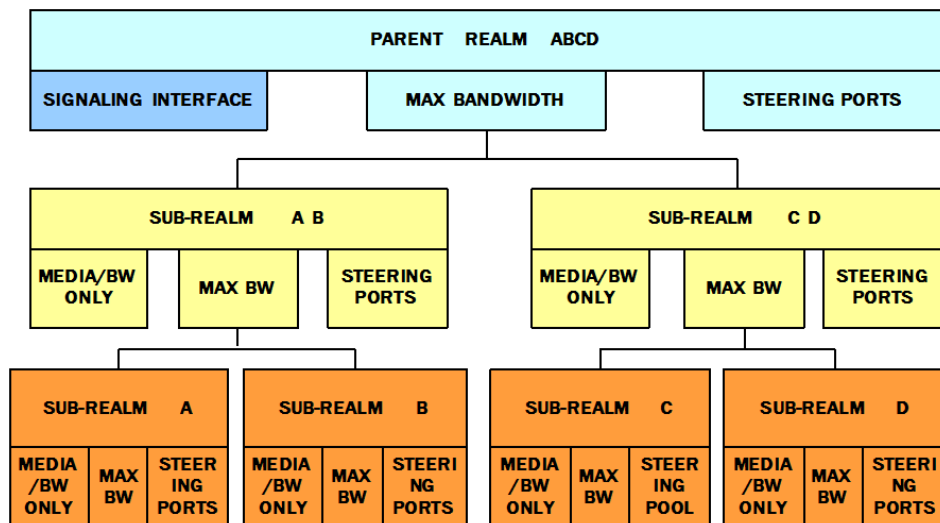
- Realm group—A hierarchical nesting of realms identified by the name of the highest order realm.
- Controlling realm—A realms for which a signaling interface is configured. For example, you might configure these signaling interfaces in the following

configurations: SIP-NAT or SIP port. Typically, this is the highest order realm for the parent realm in a realm group.

- **Parent realm**—A realm that has one or more child realms. A parent realm might also be the child realm of another realm group.
- **Child realm**—A realm that is associated with a single higher order parent realm. A child might also be the parent realm of another realm group. Child realms inherit all signaling and steering ports from higher order realms.
- **Media-only realm**—A realm for which there is no configured signaling interface directly associated. Media-only realms are nested within higher order realms.

As these definitions suggest, parent and child realms can be constructed so that there are multiple nesting levels. Lower order realms inherit the traits of the realms above them, including: signaling service interfaces, session translation tables, and steering pools.

Since realms inherit the traits of the realms above them in the hierarchy, you will probably want to map what realms should be parents and children before you start configuring them. These relationships are constructed through one parameter in the realm configuration that identifies the parent realm for the configuration. If you specify a parent realm, then the realm you are configuring becomes a child realm subject to the configured parameters you have established for that parent. And since parent realms can themselves be children of other realm, it is important that you construct these relationships with care.



Configuring Nested Realms

When you are configuring nested realms, you can separate signaling and media by setting realm parameters in the SIP interface configuration and the steering ports configuration.

- The realm identifier you set in the SIP interface configuration labels the associated realm for signaling.
- The realm identifier you set in the steering ports configuration labels the associated realm for media.

Constructing a hierarchy of nested realms requires that you note which realms you want to handle signaling, and which you want to handle media.

In the SIP port configuration for the SIP interface, you will find an allow anonymous parameter that allows you to set certain access control measures. The table below outlines what each parameter means.

Allow Anonymous Parameter	Description
all	All anonymous connections allowed.
agents-only	Connections only allowed from configured session agents.
realm-prefix	Connections only allowed from addresses with the realm's address prefix and configured session agents.
registered	Connections allowed only from session agents and registered endpoints. (For SIP only, a REGISTER is allowed for any endpoint.)
register-prefix	Connections allowed only from session agent and registered endpoints. (For SIP only, a REGISTER is allowed for session agents and a matching realm prefix.)

ACLI Instructions and Examples

To configure nested realms, you need to set parameters in the realm configuration.

To configure parent and child realms in the ACLI:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```
4. **parent-realm**—Enter the identifier of the realm you want to name as the parent. Configuring this parameter makes the realm you are currently configuring as the child of the parent you name. As such, the child realm is subject to the configured parameters for the parent.

Aggregate Session Constraints: Nested Realms

In addition to setting session constraints per realm for SIP and H.323 sessions, you can also enable the Net-Net SBC to apply session constraints across nested realms. When you set up session constraints for a realm, those constraints apply only to the realm for which they are configured without consideration for its relationship either as parent or child to any other realms.

You can also, however, enable the Net-Net SBC to take nested realms into consideration when applying constraints. For example, if a call enters on a realm that has no constraints but its parent does, then the constraints for the parent are applied. This parameter is global and so applies to all realms on the system. For the specific realm the call uses and for all of its parents, the Net-Net SNC increments the counters upon successful completion of an inbound or outbound call.

In the following example, you can see one parent realm and its multiple nested, child realms. Now consider applying these realm constraints:

- Parent Realm 1—55 active sessions
- Child Realm 1—45 active sessions
- Child Realm 2A—30 active sessions
- Child Realm 2B—90 active sessions
- Child Realm 3—20 active sessions



Given the realm constraints outlined above, consider these examples of how global session constraints for realms. For example, a call enters the Net-Net SBC on Child Realm 2B, which has an unmet 90-session constraint set. Therefore, the Net-Net SBC allows the call based on Child Realm 2B. But the call also has to be within the constraints set for Child Realm 1 and Parent Realm 1. If the call fails to fall within the constraints for either of these two realms, then the Net-Net SBC rejects the call.

On the Net-Net 9200, the maximum number of sessions applied to a realm is for each SIP interface and each H.323 stack, it is not applied as total-calls through the realm.

Impact to Other Session Constraints and Emergency Calls

You can set up session constraints in different places in your Net-Net SBC configuration. Since session agents and SIP interfaces also take session constraints, it is important to remember the order in which the Net-Net SBC applies them:

1. Session agent session constraints
2. Realm session constraints (including parent realms)
3. SIP interface session constraints

Emergency and priority calls for each of these is exempt from session constraints. That is, any call coming into the Net-Net SBC marked priority is processed.

ACLI Instructions and Examples

You enabled use of session constraints for nested realms across the entire system by setting the **nested-realms-stats** parameter in the session router configuration to **enabled**.

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **session-router** and press <Enter>.

```
ACMEPACKET(session-router)# session-router
```

```
ACMEPACKET(session-router-config)#
```

4. **nested-realms-stats**—Change this parameter from **disabled** (default) to **enabled** if you want the Net-Net SBC to apply session constraints across all nested realms (realms that are children to other realms).
5. Save and activate your configuration.

Realm-Based Packet Marking

The Net-Net 9200 supports TOS/DiffServ functions that allow you to

- Set up realm-based packet marking by media type, either audio-voice or video
- Set up realm-based packet marking for signaling for SIP

Upstream devices use these markings to classify traffic in order to determine the priority level of treatment it will receive.

About TOS/DiffServ

TOS and DiffServ are two different mechanisms used to achieve QoS in enterprise and service provider networks; they are two different ways of marking traffic to indicate its priority to upstream devices in the network.

Given the somewhat confusing differences between TOS and DiffServ (since both specify use of the same byte in the IP header), the ToS byte and DiffServ byte sections below provide some basic information for clarification.

For more information about TOS (packet) marking, refer to:

- IETF RFC 1349 (<http://www.ietf.org/rfc/rfc1349.txt>)

For more information about DiffServ, refer to:

- IETF RFC 2474 (<http://www.ietf.org/rfc/rfc2474.txt>)
- IETF RFC 2475 (<http://www.ietf.org/rfc/rfc2475.txt>).

ToS Byte

The TOS byte format is as follows:

Precedence			TOS				MBZ
0	1	2	3	4	5	6	7

The TOS byte is broken down into three components:

- **Precedence**—The most used component of the TOS byte, the precedence component is defined by three bits. There are eight possible precedence values ranging from 000 (decimal 0) through 111 (decimal 7). Generally, a precedence

value of 000 refers to the lowest priority traffic, and a precedence value of 111 refers to the highest priority traffic.

- TOS—The TOS component is defined by four bits, although these bits are rarely used.
- MBZ—The must be zero (MBZ) component of the TOS byte is never used.

DiffServ Byte

Given that the TOS byte was rarely used, the IETF redefined it and in doing so created the DiffServ byte.

The DiffServ byte format is as follows:

Precedence			TOS				MBZ
0	1	2	3	4	5	6	7

The DiffServ codepoint value is six bits long, compared to the three-bit-long TOS byte's precedence component. Given the increased bit length, DiffServ codepoints can range from 000000 (decimal 0) to 111111 (decimal 63).

Note: By default, DiffServ codepoint mappings map exactly to the precedence component priorities of the original TOS byte specification.

Packet Marking for Media

You can set the TOS/DiffServ values that define an individual type or class of service for a given realm. In addition, you can specify:

- One or more audio media types for SIP
- One or more video media types for SIP
- Both audio and video media types for SIP

For all incoming SIP and H.23 requests, the media type is determined by negotiation or by preferred codec. SIP media types are determined by the SDP.

Configuring Packet Marking by Media Type

This section describes how to set up the media policy configuration that you need for this feature, and then how to apply it to a realm.

These are the ACLI parameters that you set for the media policy:

```
name          media policy name
tos-values    list of TOS for media types for this policy
```

This is the ACLI parameter that you set for the realm:

```
media-policy default media policy name
```

ACLI Instructions and Examples

To set up a media policy configuration to mark audio-voice or video packets:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```

3. Type **media-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-policy
ACMEPACKET(media-policy)#
```

From this point, you can configure media policy parameters. To view all configuration parameters for media profiles, enter a ? at the system prompt.

4. **name**—Enter the unique name of this media policy.
5. **tos-values**—Enter the list of TOS values for media types for this media policy; by default, this list is empty. These values provide a policing profile.

Using this list, you can specify one or more audio media types, one or more video media types, or both audio and video media types.

The format for **tos-values** entry must follow: <media-type>:<tos-value>. The <media-type> portion can be either **audio** or **video**, and the <tos-value> is either a decimal or hexadecimal value to insert.

Single entries look like this:

```
ACMEPACKET(media-policy)# tos-values audio:100
```

Multiple entries for this parameter look like this:

```
ACMEPACKET(media-policy)# tos-values audio:100 video:0x33
```

You can use the **add** and **delete** commands when you enter the **tos-values** parameter to edit the list of ToS values; entering a new list of values without the **add** or **delete** commands overwrites the entire list.

```
ACMEPACKET(media-policy)# tos-values add audio:100
```

```
ACMEPACKET(media-policy)# tos-values delete audio:100
```

Applying a Media Policy to a Realm

To apply a media policy to a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```
4. **media-policy**—Enter the unique name of the media policy you want to apply to this realm.

Packet Marking for Signaling

ToS marking for signaling requires you to configure a media policy, a class profile and class policy, and then set the name of the class profile in the appropriate realm configuration.

ACLI Instructions and Examples

This section shows you how to configure packet marking for signaling.

Configuring a Media Policy for Signaling Packet Marking

To set up a media policy configuration to mark audio-voice or video packets:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# media-policy
ACMEPACKET(media-policy)#
```

From this point, you can configure media policy parameters. To view all media policy configuration parameters, enter a ? at the system prompt.
4. **name**—Enter the unique name of this media policy. When you set up the class policy, this is the value you set in the **media-policy** parameter.
5. **tos-values**—Enter the list of TOS values for media types for this media policy; by default, this list is empty. These values provide a policing profile.

Using this list, you can specify one or more audio media types, one or more video media types, or both audio and video media types.

The format for **tos-values** entry must follow: <media-type>:<tos-value>. The <media-type> portion is sip, and the <tos-value> is either a decimal or hexadecimal value to insert.

You can use the add and delete commands when you enter the tos-values parameter to edit the list of ToS values; entering a new list of values without the add or delete commands overwrites the entire list.

Configuring the Class Profile and Class Policy

To configure the class profile and class policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **class-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# class-profile
ACMEPACKET(class-profile)#
```
4. Type **policy** and press <Enter> to begin configuring the class policy.


```
ACMEPACKET(class-profile)# policy
```

From this point, you can configure class policy parameters. To view all class policy configuration parameters, enter a ? at the system prompt.
5. **profile-name**—Enter the unique name of the class policy. When you apply a class profile to a realm configuration, you use this value.
6. **to-address**—Enter a list of addresses to match to incoming traffic for marking. You can use E.164 addresses, a host domain address, or use an asterisk (*) to set all host domain addresses.

7. **media-policy**—Enter the name of the media policy you want to apply to this class policy.

Applying a Media Policy to a Realm

To apply a class policy to a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
```
4. **class-profile**—Enter the name of the class profile to apply to this realm. This is the name you set in the **profile-name** parameter of the class-policy configuration.

SIP-SDP DCSP Marking/ToS Bit Manipulation

Used to indicate priority and type of requested service to devices in the network, type of service (TOS) information is included as a set of four-bit flags in the IP header. Each bit has a different purpose, and only one bit at a time can be set: There can be no combinations. Available network services are:

- Minimum delay—Used when latency is most important
- Maximum throughput—Used when the volume of transmitted data in any period of time is important
- Maximum reliability—Used when it is important to assure that data arrives at its destination without requiring retransmission
- Minimum cost—Used when it is most important to minimize data transmission costs

The Net-Net SBC's support for type of service (TOS) allows you to base classification on the media type as well as the media subtype. In prior releases, you can configure the Net-Net SBC to mark TOS bits on outgoing packets using a media policy. Supported media types include audio, video, application, data, image, text, and message; supported protocol types are H.225, H.245, and SIP. Note that, although H.225 and H.245 are not part of any IANA types, they are special cases (special subtypes) of "message" for the Net-Net SBC. When these criteria are met for an outgoing packet, the Net-Net SBC applies the TOS settings to the IP header. The augmented application of TOS takes matching on media type or protocol and expands it to match on media type, media-sub-type, and media attributes.

The new flexibility of this feature resolves issues when, for example, a customer needs to differentiate between TV-phone and video streaming. While both TV-phone and video streaming have the attribute "media=video," TV-phone streaming has "direction=sendrcv" prioritized at a high level and video has "direction=sendonly or recvonly" with middle level priority. The Net-Net SBC can provide the appropriate marking required to differentiate the types of traffic.

How It Works

In the media policy, the **tos-values** parameter accepts values that allow you to create any media type combination allowed by IANA standards. This is a dynamic process because the Net-Net SBC generates matching criteria directly from messages.

The new configuration takes a media type value of any of these: audio, example, image, message, model, multipart, text, and video. It also takes a media sub-type of any value specified for the media type by IANA; however, support for T.38 must be entered exactly as **t.38** (rather than **t38**). Using these values, the Net-Net SBC creates a value Based on a combination of these values, the Net-Net SBC applies TOS settings.

You also configure the TOS value to be applied, and the media attributes you want to match.

You can have multiple groups of TOS settings for a media policy.

ACLI Instructions and Examples

This section provides instructions for how to configure TOS bit manipulation on your Net-Net SBC.

To configure TOS bit manipulation:

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

```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(configure)# media-manager
```

```
ACMEPACKET(media-manager)#
```

3. Type **media-policy** and press <Enter>.

```
ACMEPACKET(media-manager)# media-policy
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration you want to edit.

4. Type **tos-settings** and press <Enter>.

```
ACMEPACKET(media-policy)# tos-settings
```

5. **media-type**—Enter the media type that you want to use for this group of TOS settings. You can enter any of the IANA-defined media types for this value: audio, example, image, message, model, multipart, text, and video. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

```
ACMEPACKET(tos-settings)# media-type message
```

6. **media-sub-type**—Enter the media sub-type you want to use for the media type. This value can be any of the sub-types that IANA defines for a specific media type. This value is not case-sensitive and can be up to 255 characters in length; it has no default.

```
ACMEPACKET(tos-settings)# media-sub-type sip
```

7. **media-attributes**—Enter the media attribute that will match in the SDP. This parameter is a list, so you can enter more than one value. The values are case-sensitive and can be up to 255 characters in length. This parameter has no default.

If you enter more than one media attribute value in the list, then you must enclose your entry in quotation marks ("). Valid values are *sendonly*, *sendrecv*, and *recvonly*.

```
ACMEPACKET(tos-settings)# media-attributes "sendonly sendrecv"
```


8. **tos-values**—Enter the TOS values you want applied for matching traffic. This value is a decimal or hexadecimal value. The valid range is:
 - 0x00 to 0xFF.

```
ACMEPACKET(tos-settings)# tos-value 0xF0
```
9. Save and activate your configuration.

Steering Pools

Steering pools define sets of ports that are used for steering media flows through the Net-Net 9200. These selected ports are used to modify the SDP to cause receiving session agents to direct their media toward this Net-Net system. Media can be sent along the best quality path using these addresses and ports instead of traversing the shortest path or the BGP-4 path.

For example, when the Net-Net 9200 is communicating with a SIP device in a specific realm defined by a steering pool, it uses the IP address and port number from the steering pool's range of ports to direct the media. The port the Net-Net 9200 chooses to use is identified in the SDP part of the message.

Note: The values entered in the steering pool are used when the Net-Net system provides NAT, PAT, and VLAN translation.

Configuration Overview

To plan steering pool ranges, take into account the total sessions available on the box, determine how many ports these sessions will use per media stream, and assign that number of ports to all of the steering pools on your Net-Net 9200. For example, if your Net-Net 9200 can accommodate 500 sessions and each session typically uses 2 ports, you would assign 1000 ports to each steering pool. This strategy provides for a maximum number of ports for potential use, without using extra resources on ports your Net-Net 9200 will never use.

The following table lists the steering pool parameters you need to configure:

Parameter	Description
IP address	IP address of the steering pool.
start port	Port number that begins the range of ports available to the steering pool. You must define this port to enable the Net-Net system to perform media steering and NATing.
end port	Port number that ends the range of ports available to the steering pool. You must define this port to enable the Net-Net system to perform media steering and NATing.
realm id	Identifies the steering pool's realm. The steering pool is restricted to only the flows that originate from this realm.

Note: The combination of entries for IP address, start port, and realm ID must be unique in each steering pool. You cannot use the same values for multiple steering pools.

Each bidirectional media stream in a session uses two steering ports, one in each realm (with the exception of audio/video calls that consume four ports). You can

configure the start and end port values to provide admission control. If all of the ports in all of the steering pools defined for a given realm are in use, no additional flows/sessions can be established to/from the realm of the steering pool.

ACLI Instructions and Examples

The following example shows the configuration of a steering pool.

```
steering-pool
  ip-address          192.168.0.11
  start-port          20000
  end-port            21000
  realm-id            peer-1
last-modified-date    2005-03-04 00:35:22
```

To configure a steering pool:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **steering-pool** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# steering-pool
ACMEPACKET(steering-pool)#
```
4. **ip-address**—Enter the target IP address of the steering pool in IP address format. For example, 192.168.0.11.
5. **start-port**—Enter the start port value that begins the range of ports available to this steering pool. The available values include:
 - minimum (default): 0 (zero)
 - maximum: 65535

You must enter a valid port number or the steering pool will not function properly.
6. **end-port**—Enter the end port value that ends the range of ports available to this steering pool. The available values include:
 - minimum (default): 0 (zero)
 - maximum: 65535

You must enter a valid port number or the steering pool will not function properly.
7. **realm-id**—Enter the realm ID to identify the steering pool's realm, following the name format. The value you enter here must correspond to the value you entered as the identifier (name of the realm) when you configured the realm. For example, peer-1.

This steering pool is restricted to flows that originate from this realm.

Multiple Interface Realms

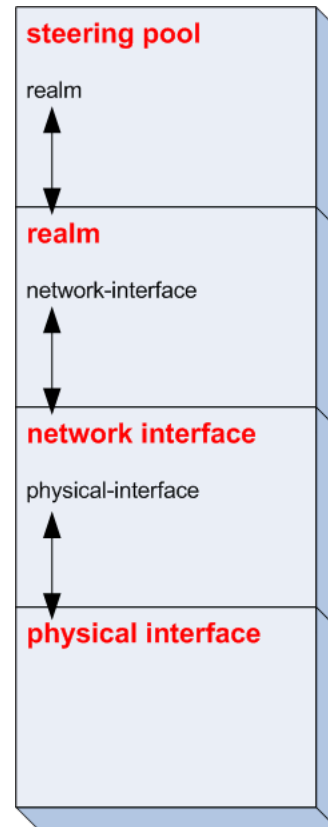
The multi-interface realm feature lets you group multiple network interfaces to aggregate their bandwidth for media flows. In effect, this feature lets you use the total throughput of the available physical interfaces on your Net-Net 9200 for a

single realm. Multi-interface realms are implemented by creating multiple steering pools, each on an individual network interface, that all reference a single realm.

Of course, you can not to use this feature and configure your Net-Net 9200 to create a standard one-realm to one-network interface configuration.

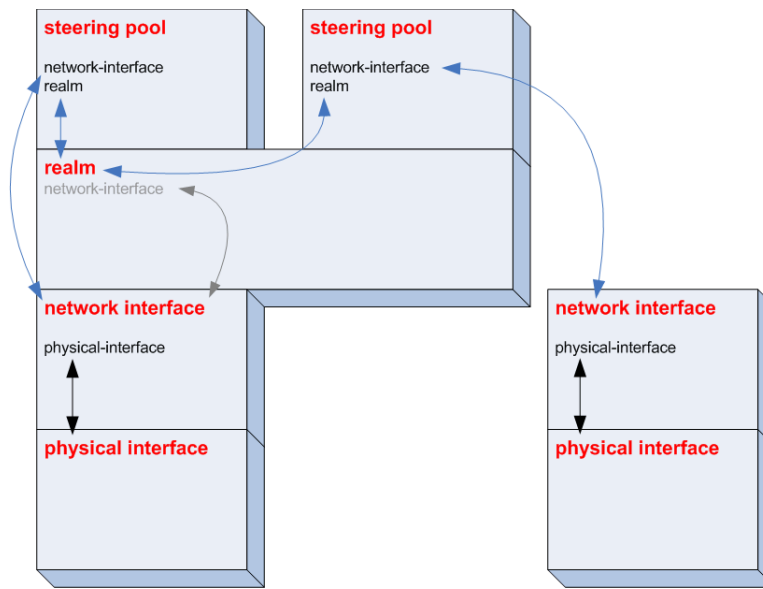
How It Works

Without using multiple interface realms, the basic hierarchical configuration of the Net-Net 9200 from the physical interface through the media steering pool looks like this:

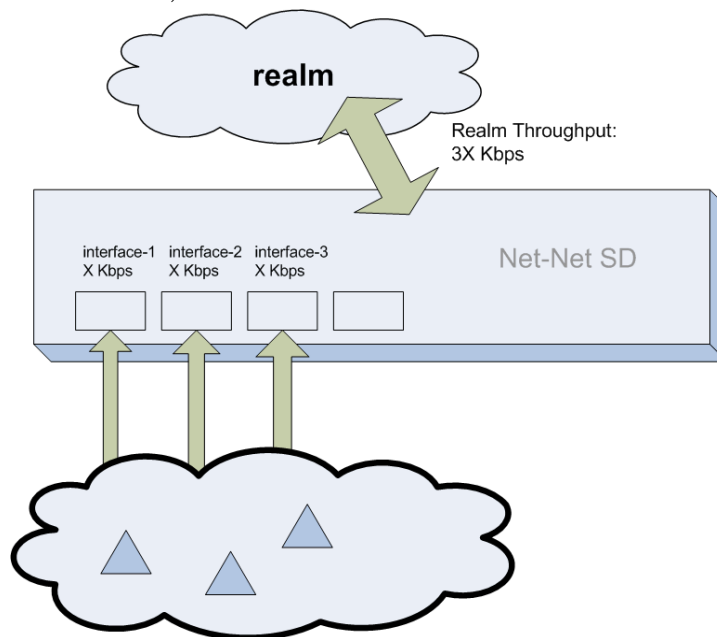


In this model, one (non-channelized) network interface exists on a physical interface. One realm exists on one network interface. One or more steering pools can exist on one realm. Within each higher level configuration element exists a parameter that references a lower level configuration element in the Net-Net 9200's logical network model.

The multi-interface realm feature directs media traffic entering and exiting multiple network interfaces in and out of a single realm. Since all the steering pools belong to the same realm, their assigned network interfaces all feed into the same realm as well. The following diagram shows the relationship in the new logical model:



The advantage of using multi-interface realms is the ability to aggregate the bandwidth available to multiple network interfaces for a larger-than-previously-available total bandwidth for a realm. In the illustration below, three physical interfaces each have X Kbps of bandwidth. The total bandwidth available to the realm with multiple network interfaces is now 3X the bandwidth. (In practical usage, interface-1 only contributes X - *VoIP Signaling* to the total media bandwidth available into the realm.)



Steering Pool Port Allocation

Every steering pool you create includes its own range of ports for media flows. The total number of ports in all the steering pools that feed into one realm are available for calls in and out of the realm.

Steering pool ports for a given realm are assigned to media flows sequentially. When the first call enters the Net-Net 9200 after start-up, it is assigned the first ports on the first steering pool that you configured. New calls are assigned to ports sequentially in the first steering pool. When all ports from the first steering pool are exhausted, the Net-Net 9200 uses ports from the next configured steering pool. This continues until the last port on the last configured steering pool is used.

After the final port is used for the first time, the next port chosen is the one first returned as empty from the full list of ports in all the steering pools. As media flows are terminated, the ports they used are returned to the realm's full steering pool. In this way, after initially exhausting all ports, the realm takes new, returned, ports from the pool in a "least last used" manner.

When a call enters the Net-Net 9200, the signaling application allocates a port from all of the eligible steering pools that will be used for the call. Once a port is chosen, the Net-Net 9200 checks if the steering pool that the port is from has a defined network interface. If it does, the call is set up on the corresponding network interface. If a network interface is not defined for that steering pool, the network interface defined for the realm is used.

ACLI Instructions and Examples

This section explains how to configure your Net-Net 9200 to use multiple interface realms.

Creating a List of Network Interfaces for the Realm

You must first configure multiple physical interfaces and multiple network interfaces on your Net-Net 9200.

To configure the realm configuration for multi-interface realms.

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-manager path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

From this point, you can configure a realm that will span multiple network interfaces.
4. **network-interfaces**—Enter the name of the network interface where the signaling traffic for this realm will be received.

Creating Steering Pools for Multiple Interface Realms

To configure steering pools for multi-interface realms:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-manager path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **steering-pool** and press <Enter>. The system prompt changes.


```
ACMEPACKET(media-manager)# steering-pool
ACMEPACKET(steering-pool)#
```

From this point, you can configure steering pools which collectively bridge the multiple network interfaces they are connected to.

4. **ip-address**—Enter the IP address of the first steering pool on the first network interface.

This IP address must correspond to an IP address within the subnet of a network interface you have already configured.

This IP can not exist on a network interface other than the one you configure in the **network-interface** parameter.
5. **start-port**—Enter the beginning port number of the port range for this steering pool.
6. **end-port**—Enter the ending port number of the port range for this steering pool.
7. **realm-id**—Enter the name of the realm which this steering pool directs its media traffic toward.
8. **network-interface**—Enter the name of the network interface you want this steering pool to direct its media toward. This parameter will match a **name** parameter in the network interface configuration element. If you do not configure this parameter, you can only assign a realm to a single network interface.
9. Create additional steering pools on this and on other network interfaces as needed. Remember to type **done** when you are finished configuring each new steering pool.

Media over TCP

The Net-Net 9200 supports RFC 4145 (TCP-Based Media Transport in the SDP), also called TCP Bearer support. Media over TCP can be used to support applications that use TCP for bearer path transport.

RFC 4145 adds two new attributes, *setup* and *connection*, to SDP messages. The *setup* attribute indicates which end of the TCP connection should initiate the connection. The *connection* attribute indicates whether an existing TCP connection should be used or if a new TCP connection should be setup during re-negotiation. RFC 4145 follows the offer/answer model specified in RFC3264. An example of the SDP offer message from the end point 192.0.2.2 as per RFC4145 is as given below:

```
m=image 54111 TCP t38
c=IN IP4 192.0.2.2
a=setup:passive
a=connection:new
```

This offer message indicates the availability of t38 fax session at port 54111 which runs over TCP. Net-Net 9200 does not take an active part in the application-layer communication between each endpoint.

The Net-Net 9200 provides the means to set up the end-to-end TCP flow by creating the TCP/IP path based on the information learned in the SDP offer/answer process.

TCP Bearer Conditions

The following conditions are applicable to the Net-Net 9200's support of RFC 4145.

1. The Net-Net 9200 can not provide media-over-TCP for HNT scenarios (endpoints behind NATs).

2. If media is released into the network, the TCP packets do not traverse the Net-Net. Therefore, no TCP bearer connection is created.
3. The Net-Net 9200 does not inspect the *setup* and *connection* attributes in the SDP message since the TCP packets transparently pass through the Net-Net 9200. These SDP attributes are forwarded to the other endpoint. It is the other endpoint's responsibility to act accordingly.
4. After the Net-Net 9200 receives a SYN packet, it acts as a pure pass through for that TCP connection and ignores all further TCP handshake messages including FIN and RST. The flow will only be torn down in the following instances:
 - The TCP initial guard timer, TCP subsequent guard timer, or the TCP flow time limit timer expire for that flow.
 - The whole SIP session is torn down.

TCP Port Selection

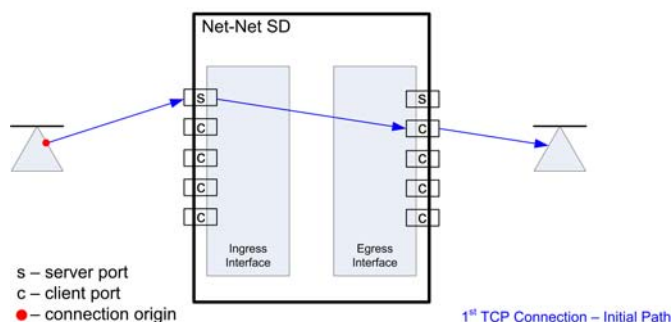
When a call is first set up, the Net-Net 9200 inspects the SDP message's m-line to see if any media will be transported via TCP. If the SDP message indicates that some content will use TCP, the Net-Net 9200 allocates a configured number of steering ports for the media-over-TCP traffic. These TCP media ports are taken from the each realm's steering pool.

Each endpoint can initiate up to four end-to-end TCP flows between itself and the other endpoint. The Net-Net 9200 assigns one port to receive the initial TCP packet (server port), and one to four ports assigned to send TCP traffic (client ports) to the receiving side of the TCP flow. The number of TCP flows for each call is configured globally.

In order to configure the Net-Net 9200 to facilitate and support this process, you need to specify the number of ports per side of the call that can transport discrete TCP flows. You can configure one to four ports/flows. For configuration purposes, the Net-Net 9200 counts this number as inclusive of the server port. Therefore if you want the Net-Net 9200 to provide a maximum of one end-to-end TCP flow, you have to configure two TCP ports; one to receive, and one to send. The receiving port (server) is reused to set up every flow, but the sending port (client) is discrete per flow. For example: for 2 flows in each direction, set the configuration to 3 TCP ports per flow; for 3 flows in each direction, set the configuration to 4 TCP ports per flow, etc.

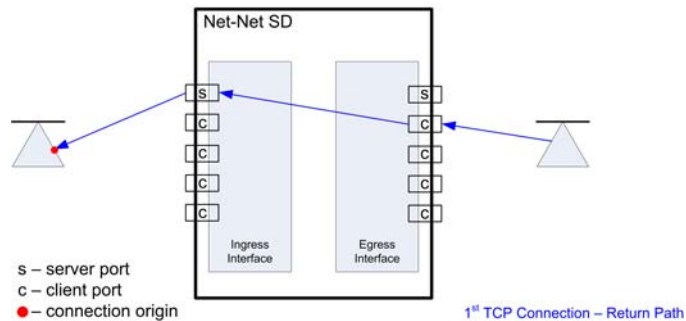
The server port is used for initiating a new TCP connection. An endpoint sends the first packet to a server port on the ingress interface. The packet is forwarded out of the Net-Net 9200 through a client port on the egress interface toward an endpoint:

TCP Connection 1 - Eastward Path



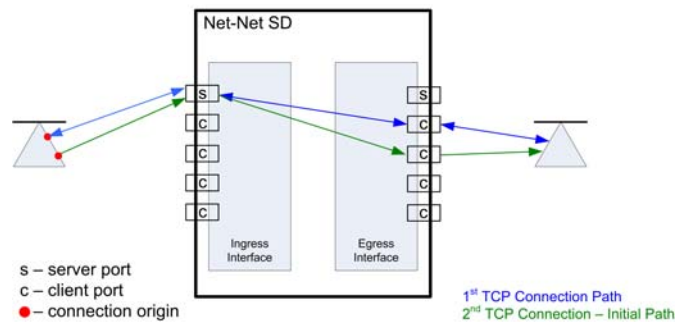
The endpoint responds back to the client port on the egress interface. This message traverses the Net-Net 9200 and is forwarded out of the server port on the ingress interface where the initial packet was sent. The remainder of the TCP flow uses the server and client port pair as a tunnel through the Net-Net 9200:

TCP Connection 1 - Westward Path



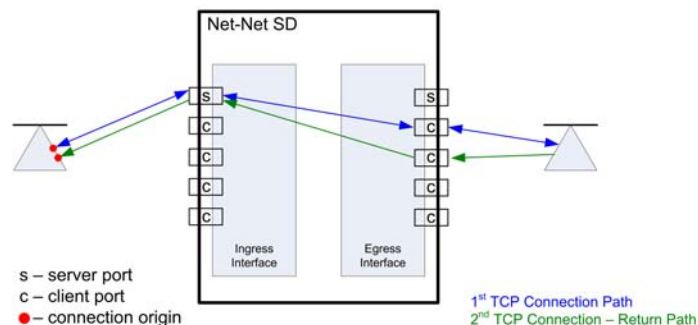
When the second TCP connection is set up in the same direction as in the first example, the first packet is still received on the server port of the ingress interface. The next unused client port is chosen for the packet to exit the Net-Net 9200:

TCP Connection 2 - Eastward Path



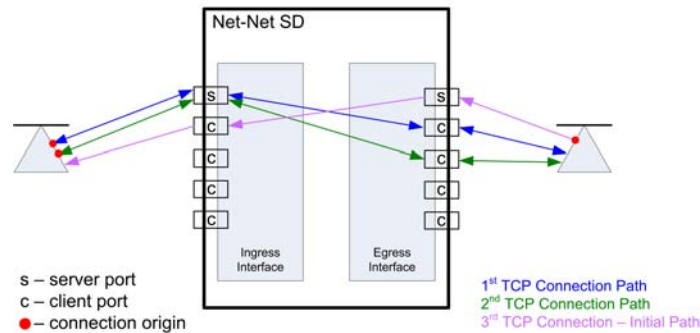
The response takes the same path back to the caller. The remainder of the second TCP connection uses this established path:

TCP Connection 2 - Westward Path



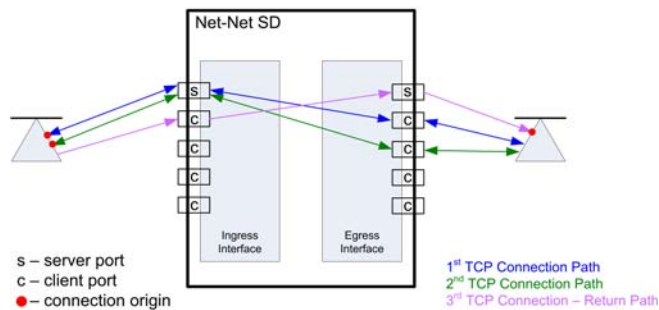
When the callee initiates a TCP connection, it must send its initial traffic to the server port on its Net-Net 9200 ingress interface. The packet is forwarded out of the first free client port on the egress side of this TCP connection toward the caller.

TCP Connection 3 – Callee Initiates Connection



The caller's response takes the same path back to the callee that initiated this TCP connection. The remainder of the third TCP connection uses this established path.

TCP Connection 3 – Return Path: Caller to Callee

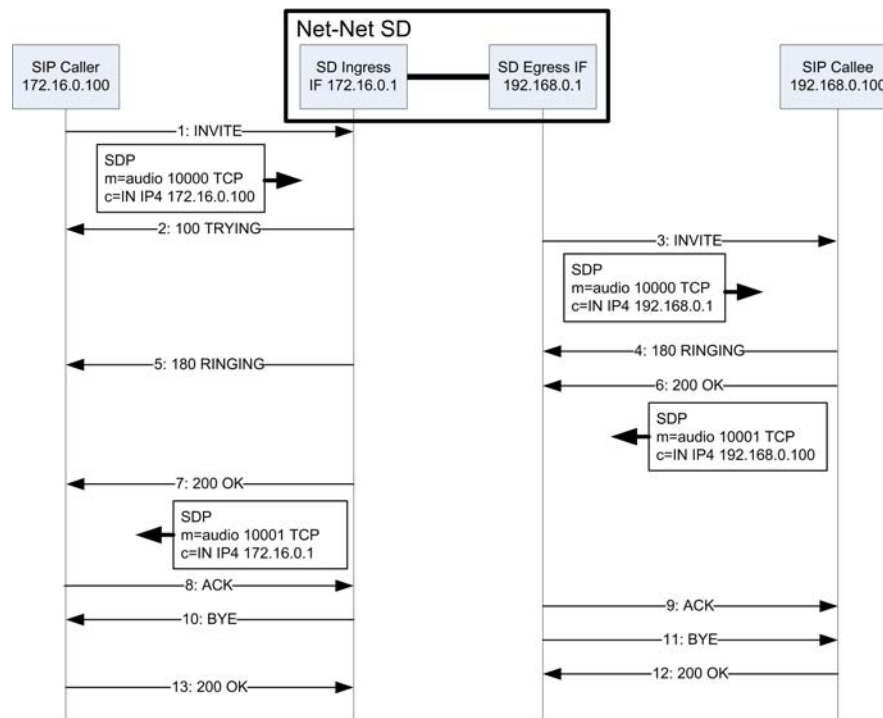


The Net-Net 9200 can support a total of eight media-over-TCP connections per call. A maximum of 4 connections are supported as initiated from each side of the call.

SDP Offer Example

The following abbreviated call flow diagram sets up a media-over-TCP flow. Observe that the caller listens for audio over TCP on 172.16.0.10:10000, as described in the SDP offer (1). The Net-Net 9200 re-writes the m and c lines in the SDP offer to reflect that it is listening for audio over TCP on its egress interface at 192.168.0.1:10000 (3). The Net-Net 9200 then forwards the SIP invite to the callee.

The SIP callee responds with an SDP answer in a 200 OK message. The callee indicates it is listening for the audio over TCP media on 192.168.0.10:10001 (6). The Net-Net 9200 re-writes the m and c lines in the SDP answer to reflect that it is listening for audio over TCP on the call's ingress interface at 172.16.0.1:10001 (7). The Net-Net 9200 then forwards the SIP invite to the caller.



All interfaces involved with the end-to-end TCP flow have now established their listening IP address and port pairs.

Timers

The Net-Net 9200 has three guard timers that ensure a TCP media flow does not remain connected longer than configured. You can set each of these from 0 (disabled) to 999999999 in seconds.

- TCP initial guard timer — Sets the maximum time in seconds allowed to elapse between the initial SYN packet and the next packet in this flow.
- TCP subsequent guard timer — Sets the maximum time in seconds allowed to elapse between all subsequent sequential TCP packets.
- TCP flow time limit — Sets the maximum time that a single TCP flow can last. This does not refer to the entire call.

ACLI Instructions and Examples

To configure media over TCP:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-manager** and press <Enter> to begin configuring media over TCP.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. **tcp-number-of-ports-per-flow**—Enter the number of port to use.
5. **tcp-initial-guard-timer**—Enter the value in seconds for the initial guard timer for TCP.
6. **tcp-subsq-guard-timer**—Enter the value in seconds for the subsequent guard timer for TCP.
7. **tcp-flow-time-limit**—Enter the value in seconds for the total flow time limit for TCP.

Media Release Behind the Same IP Address

The media management behind the same IP feature lets the Net-Net SBC release media when two endpoints are behind the same IP address, in the same realm. Using this feature allows intra-site call media to bypass traversing the Net-Net SBC. You can use this feature for both hosted NAT traversal (HNT) and non-HNT clients, or a combination thereof, sharing the same public IP address.

Additional Media Management Options

Additional media management options include:

- Media directed between sources and destinations within this realm on this specific Net-Net SBC. Media travels through the Net-Net SBC rather than straight between the endpoints. (**mm-in-realm** parameter)
- Media directed through the Net-Net SBC between endpoints that are in different realms, but share the same subnet. (**mm-in-network** parameter)
- For SIP only, media released between multiple Net-Net SBCs. (**msm-release** parameter)

To enable SIP distributed media release, you must set the appropriate parameter in the realm configuration. You must also set the SIP options parameter to media-release with the appropriate header name and header parameter information. This option defines how the Net-Net SBC encodes IPv4 address and port information for media streams described by, for example, SDP.

Configuring Media Release Behind the Same IP Address

You need to configure both the mm-in-realm and mm-same-ip parameters for the realm:

- If the mm-in-realm parameter is disabled, the mm-same-ip parameter is ignored.
- If the mm-in-realm parameter is enabled and the mm-same-ip parameter is disabled, media will be managed in the realm but released if the two endpoints are behind the same IP address.

ACLI Instructions and Examples

To configure media management:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
```

```
ACMEPACKET(realm-config)#
```

From this point, you can configure realm parameters. To view all realm configuration parameters, enter a **?** at the system prompt.

4. **mm-in-realm**—Ensure this parameter is set to **enabled** if you plan to use mm-same-ip. If set to **disabled**, the mm-same-ip parameter is ignored. If you set this to **enabled** and mm-same-ip to **disabled**, media will be managed in the realm but released if the two endpoints are behind the same IP address
5. **mm-same-ip**—Retain the default of **enabled** if you want media to go through this Net-Net SBC. If set to **disabled**, the media will not go through the Net-Net SBC for endpoints that are behind the same IP.

Bandwidth CAC for Media Release

The bandwidth CAC for media release feature adds per-realm configuration that determines whether or not to include inter-realm calls in bandwidth calculations. When you use this feature, the Net-Net SBC's behavior is to count and subtract bandwidth from the used bandwidth for a realm when a call within a single site has its media released. When you do not enable this feature (and the Net-Net SBC's previous behavior), the Net-Net does not subtract the amount of bandwidth.

In other words:

- When you enable this feature, an inter-realm media-released call will decrement the maximum bandwidth allowed in that realm with the bandwidth used for that call.
- When you disable this feature (default behavior), and inter-realm media-released call will not decrement the maximum bandwidth allowed for that call.

ACLI Instructions and Examples

To enable bandwidth CAC for media release:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Select the realm where you want to add this feature.


```
ACMEPACKET(realm-config)# select
```
5. **bw-cac-non-mm**—Enable this parameter to turn on bandwidth CAC for media release. The default is **disabled**. The valid values are:
 - enabled | disabled
6. Save and activate your configuration.

Media Release between Endpoints with the Same IP Address

You can configure your Net-Net SBC to release media between two endpoints even when one:

- Is directly addressable at the same IP address as a NAT device, but is not behind a NAT device
- The other is behind a NAT device with the same address

You enable this feature on a per-realm basis by setting an option in the realm configuration.

When this option is not set, the Net-Net SBC will (when configured to do so) release media between two endpoints sharing one NAT IP address in the same realm or network.

ACLI Instructions and Examples

In order for this feature to work properly, the following conditions apply for the realm configuration:

- Either the **mm-in-realm** or the **mm-in-network** parameter must be disabled; you can have one of these enabled as long as the other is not. The new option will apply to the parameter that is disabled.
- If either the **mm-in-realm** or **mm-in-network** parameter is enabled, then the **mm-same-ip** parameter must be disabled.

To enable media release between endpoints with the same IP address:

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

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

2. Type **media-router** and press <Enter>.

```
ACMEPACKET(configure)# media-router
ACMEPACKET(media-router)#
```

3. Type **realm-config** and press <Enter>.

```
ACMEPACKET(media-router)# realm-config
```

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **release-media-at-same-nat** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Media Release Behind the Same NAT IP Address

You can now configure your Net-Net SBC to release media between endpoints sharing the same NAT IP address, even if one endpoint is at—but not behind—the same NAT. This feature expands on the Net-Net SBC’S pre-existing ability to release media between calling and called parties behind the same IP address/NAT device in the same realm or network.

ACLI Instructions and Examples

For this feature to work properly, your realm configuration should either have the **mm-in-realm** or **mm-in-network** parameter set to **disabled**, unless the **mm-same-ip** parameter is set to **disabled**. If the **mm-same-ip** parameter is **disabled**, then **mm-in-realm** or **mm-in-network** can both be enabled.

To set the option that enables media release behind the same IP address:

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

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

2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type **realm-config** and press <Enter>.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **release-media-at-same-nat** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Media Profiles Per Realm

For different codecs and media types, you can set up customized media profiles that serve the following purposes:

- Police media values
- Define media bandwidth policies

You can use media profiles globally for the Net-Net SBC, or you can configure them for application on a per-realm basis. For a realm, you can configure a list of media profiles you want applied. The Net-Net SBC matches the value you set for the **match-media-profiles** parameter, and then applies those media profiles to the realm itself and to all of its child realms (but not to its parent realms).

Note: This feature has no impact on the ways the Net-Net SBC uses media profiles non-realm applications such as: H.323 interfaces, SIP interfaces, IWF, session agents, codec policies, and policy attributes.

Call Admission Control and Policing

The Net-Net SBC supports call admission control (CAC) based on realm, and it applies the limits on either ingress or egress bandwidth counters. If a calls exceeds bandwidth on either the ingress or egress side, the Net-Net SBC rejects the call. You can also use per-user CAC, which applies the maximum bandwidth from all negotiated codecs among both sides of the call.

When you apply media profiles to a realm, the Net-Net SBC applies bandwidth policing from the flow's ingress realm media profile. In the diagram below, the Net-Net SBC policies traffic for Realm A based Realm A's policing values, and the same is true for Realm B.



ACLI Instructions and Examples

This section shows you how to configure multiple media profiles per realm, and it explains how to use wildcarding.

To reference a media profile in this list, you need to enter its name and subname values in the following format `<name>::<subname>`.

About Wildcarding

You can wildcard the name portion of the media profile to provide a specific subname that the Net-Net SBC uses to find matching media profiles.

You can also enter the name value on its own, or wildcard the entire value. Leaving the subname value empty is also significant in that it allows the realm to use all media profile that have no specified **subname**. However, you cannot leave the **name** portion of the value unspecified (as all media profiles are required to have names).

Consider the examples in the following table:

Syntax	Example Value	Description
<code><name></code>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <code><name>::*</code> .
<code><name>::</code>	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<code><name>::<subname></code>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
<code>*</code>	<code>*</code>	Matches anything, but does not have to be a defined media profile.
<code>*::<subname></code>	<code>*::64k</code>	Matches all media profiles with this subname. You might have a group of media profiles with different names, but the same subname value.

Syntax	Example Value	Description
*::	*::	Matches any media profiles with an empty subname parameter.
::	::	Invalid
::*	::*	Invalid

The Net-Net SBC performs matching for wildcarded **match-media-profiles** values last. Specific entries are applied first and take precedence. When the Net-Net SBC must decide between media profiles matches, it selects the first match.

To use media profiles for a realm:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **match-media-profiles**—In the form **<name>::<subname>** or other combinations from the above table. Enter the media profiles you would like applied to this realm. These values correspond to the name and subname parameters in the media profile configuration. You can wildcard the subname parameter, or you can leave the <subname> portion empty. Refer to the [About Wildcarding \(119\)](#) section above for more information about syntax, wildcarding, and empty values.
This parameter has no default.
5. Save and activate your configuration.

Peer to Peer MSRP TCP Stitching

MSRP is used for messaging, presence, and file share in SIP over 3G services, including and most notably with RCS (Rich Communications Suite).

Mobile service providers who deploy RCS will utilize MSRP with either an MSRP application server, or in peer to peer mode. In P2P MSRP, SIP and the SDP offer / answer is used to establish a TCP media bearer plane between two MSRP endpoints. This TCP bearer plane is anchored at the SBC, is used to transport the MSRP protocol itself between the endpoints.

The Net-Net SBC supports peer-to-peer TCP connections for peers behind NATs, enabling Message Session Relay Protocol (MSRP) client to communicate with one another. More specifically, the Net-Net SBC can:

- Establish incoming TCP connections with each endpoint participating in the MSRP session using a 3-way handshake. The Net-Net SBC receives incoming SYNs on the local address and port provided in the SDP offer and answer to each endpoint.
- Stitch together the two TCP connections internally after successful establishment of both connections. This capability is used when the caller and

the callee initiate TCP SYNs towards one another via the Net-Net SBC; the “stitching” makes both clients think they are talking to a server. To achieve this end, the Net-Net SBC caches SYNs from both sides so it can modify the SYN packets to SYN-Acks with the correct sequence and Ack numbers.

Note, though this case is rare, that if a user is behind a NAT offers a=passive, then this feature cannot function properly.

- Relay MSRP stream between the endpoints.
- Police bandwidth for MSRP streams based on a defined media profile for MSRP.

ACLI Instructions and Examples

This feature is enabled automatically. You do not need to take configuration steps to use it.

Introduction

This chapter explains how to configure the Net-Net 9200 to support Session Initiation Protocol (SIP) signaling services for hosted IP services applications. SIP is a text-based application-layer signaling protocol that creates, identifies, and terminates multimedia sessions between devices.

About the Net-Net 9200 and SIP

This section describes the Net-Net 9200's support of SIP. It provides the basic information you need to understand before you configure the Net-Net 9200 for SIP signaling.

Types of SIP Devices

There are four types of SIP devices:

- SIP user agent (UA) is an endpoint in SIP end-to-end communication. A UA is a user agent client (UAC) when it initiates a request and waits to receive a response. A UA is a user agent server (UAS) when it receives a request and generates a response. A given UA will be a UAC or a UAS depending on whether it is initiating the request or receiving the request.
- A SIP proxy (or proxy server) is an intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server's primary role is routing. Its job is to ensure that a request is sent to another entity closer to the targeted user. A proxy interprets, and if necessary, rewrites specific parts of a request message before forwarding it.
- A SIP redirect server is a UAS that generates redirect responses to requests it receives, directing the client to contact an alternate set of targets. Unlike a proxy which forwards the request to the alternate set of targets, the redirect response tells the UAC to directly contact the alternate targets.
- A SIP registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles. Proxies and redirect servers can use the information from the location service to determine the location of the targeted user.

A redirect server and a registrar are each a special type of UA because they act as the UAS for the requests they process.

Basic Service Models

The Net-Net 9200 operates as a back-to-back user agent (B2BUA) within the following two basic service models:

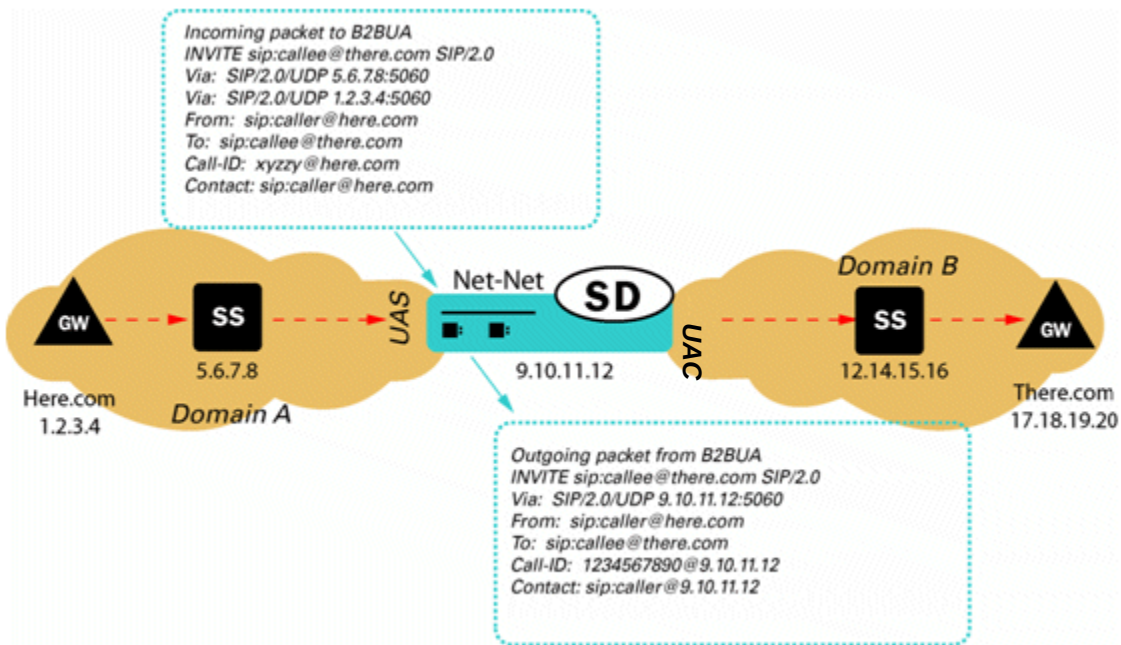
- peering
- hosted IP services

About B2BUA

A B2BUA is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests. It maintains dialog state and must participate in all requests sent on the dialogs it has established.

SIP B2BUA Peering

The Net-Net 9200 operates as a SIP B2BUA. It terminates SIP sessions and re-originates them as new sessions as they are routed through the Net-Net 9200. For each session, it establishes NATP translations and re-writes SDP to allow all session related media to be routed through the Net-Net 9200. It generates new call IDs and modifies SIP headers to prevent any protected SIP addresses and route information from being transmitted to external peers. The Net-Net 9200 supports multiple SIP interfaces that are associated with a set of media ports, thus appearing as multiple virtual SIP gateways.

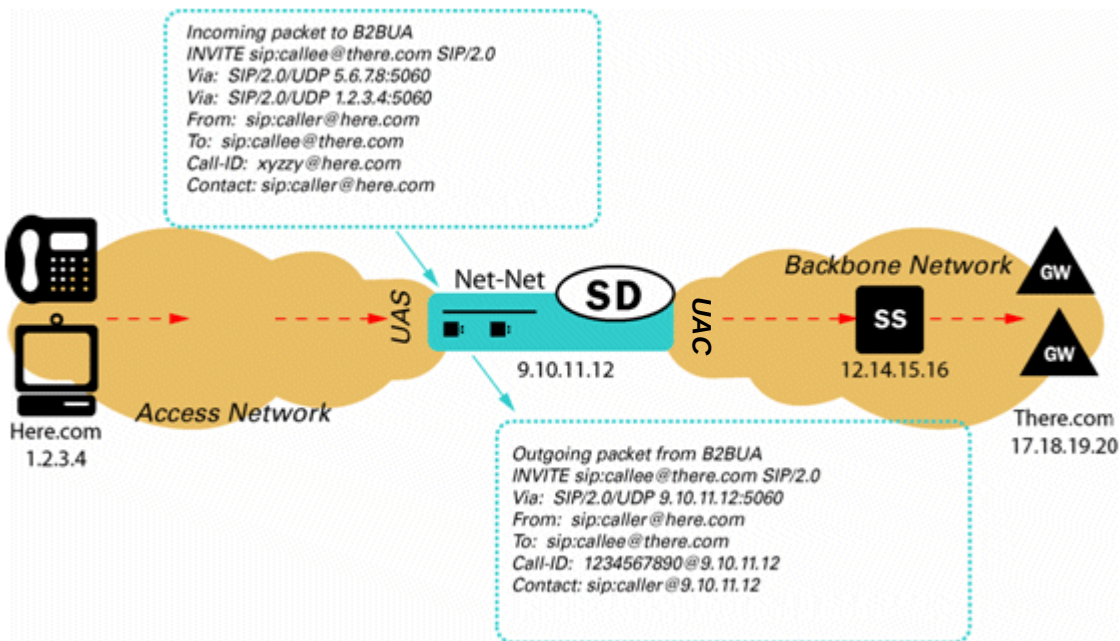


B2BUA Hosted IP Services

The Net-Net 9200 acts as an outbound proxy for SIP endpoints and performs the operations required to allow UAs behind NATs to initiate and terminate SIP sessions (Hosted NAT Traversal).

The Net-Net 9200 caches registration requests from SIP endpoints and forwards them to the appropriate softswitch or registrar in its backbone network. All subsequent signaling between the endpoint and the backbone network is through

the Net-Net 9200. Also, all calling features such as caller ID, call waiting, three-way calling, and call transfer are all supported transparently through the Net-Net 9200.



SIP B2BUA and L3/L5 NAT

For each SIP session, the Net-Net 9200 establishes NAPT translations and re-writes SDP to route all session related media through the Net-Net 9200. These actions make the Net-Net 9200 look like a SIP gateway. Also, the Net-Net 9200 support of multiple SIP interfaces associated with different network interfaces makes it appear as multiple virtual SIP gateways.

This functionality enables the Net-Net 9200 to deliver VoIP services to multiple end users, across a VPN backbone.

About SIP Interfaces

The SIP interface defines the transport addresses (IP address and port) upon which the Net-Net 9200 receives and sends SIP messages. You can define a SIP interface for each network or realm to which the Net-Net 9200 is connected. SIP interfaces support both UDP and TCP transport, as well as multiple SIP ports (transport addresses). The SIP interface's SIP NAT function lets Hosted NAT Traversal (HNT) be used in any realm.

SIP Network Models and Types

In this feature, the registration caching and SIP transport modules have been enhanced to support more SIP registrations and endpoints. Network model and network type parameters have been added for configuring this feature.

Network Model

The network model is configured in the SIP config. It indicates and optimizes for whether the SBC is deployed in **access** or **peering** scenario. There is also a **shared** network model for deployments that fall between access and peering scenarios.

Network Type

The network type is configured on each SIP interface, based upon the intended function of that SIP interface. The network type can be configured as **access**, used for the access side of a SIP deployment, or **core**, used for the core network side of a SIP deployment. In addition, the network type can be set to **any**, that will function in both roles.

Default Network Model and Type

If you have not explicitly configured the network model, it is determined by examining the configuration for the SIP Interfaces. If any enabled SIP Interface has registration caching or NAT traversal enabled, the access network model is used. If there are no SIP Interfaces with registration caching or NAT traversal enabled, the peering model is used.

If you have not explicitly configured the SIP interface's network type, it is determined based on the registration cache and NAT traversal configuration. When registration caching or NAT traversal are enabled on a SIP interface, the access network type is used. Otherwise, the core network type is used.

Upgrade Considerations

The network model and network type parameters are dynamically configurable, however, they are not made until the SPU transitions from active to standby or vice versa. Therefore, a manual switchover (or reboot) of the SPU is required after the configuration is activated in order for the new network model and type assignments to take affect. Hitless upgrades from a prior release are not supported. A full reboot of the Net-Net 9200 is required. Please contact TAC (tac@acmepacket.com) for more information.

ACLI Instructions and Examples

This section explains how to configure the network type and network model on the Net-Net 9200.

Network Model

To configure the network model in the SIP config:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
```
4. **network-model**—Set this parameter to either **access**, **peering**, or **shared** based upon the function of this Net-Net 9200 in your SIP deployment.
5. Save and activate your configuration.

Network Type

To configure the network type on a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press <Enter>.

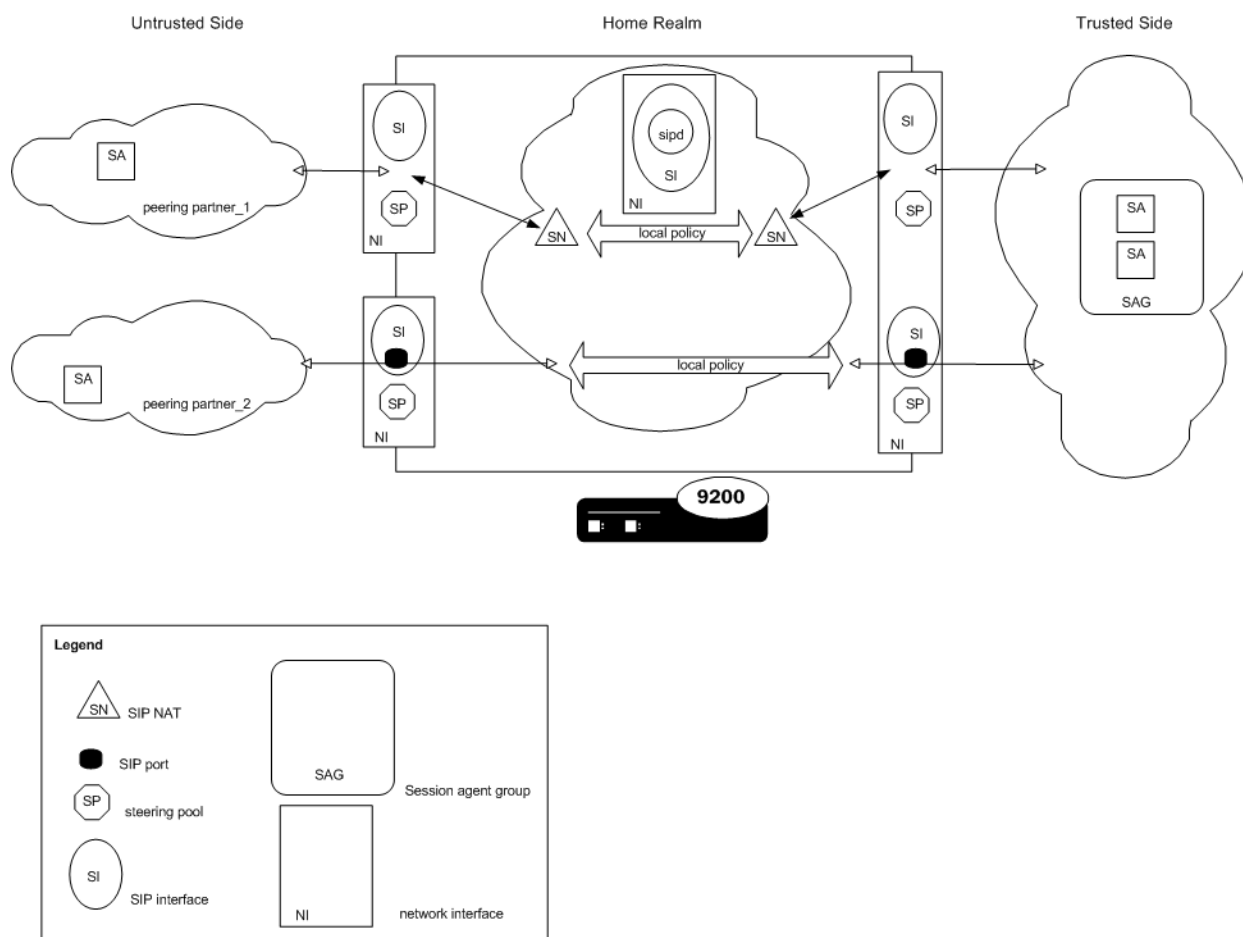
ACMEPACKET(session-router)# **sip-interface**

If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI **select** command) the SIP interface that you want to edit.

4. **network-type**—Set this parameter to either **access**, **core**, or **any** based upon the function of this SIP interface.
5. Save and activate your configuration.

The Net-Net 9200 and SIP Signaling

This section contains a diagram of a B2BUA peering environment that illustrates the Net-Net 9200 components you need to configure.



Refer to the following sections for details about configuring the Net-Net 9200 for SIP signaling.

The Home Realm

This section explains how to configure a home realm. The home realm applies only to a SIP configuration. It represents the internal default realm or network for the Net-Net 9200 and is where the Net-Net 9200's SIP proxy is located.

Overview

You primarily use a home realm when using the SIP NAT function to connect multiple realms/networks to the Net-Net 9200. You define the home realm defined as either public or private for the purposes of using the SIP NAT function. (See [The SIP NAT Function \(154\)](#) for more information). If the home realm is public, all external realms are considered private. If the home realm is private, all external networks are considered public. Usually the home realm is public.

Messages are encoded (for example, the topology is hidden) when they pass from a private to a public realm. Messages are decoded when they pass from a public realm to a private realm.

These external realms/networks might have overlapping address spaces. Because SIP messages contain IP addresses, but no layer 2 identification (such as a VLAN tag), the SIP proxy must use a single global address space to prevent confusing duplicate IP addresses in SIP URIs from different realms.

SIP NAT Function

The SIP NAT function converts external addresses in SIP URIs to an internal home realm address. Usually the external address is encoded into a cookie that is added to the userinfo portion of the URI and the external address is replaced with a home realm address unique to the SIP NAT (the SIP NAT home address).

URIs are encoded when they pass from a private realm to a public realm. When an encoded URI passes back to the realm where it originated, it is decoded (the original userinfo and host address are restored). The encoding/decoding process prevents the confusion of duplicate addresses from overlapping private addresses. It can also be used to hide the private address when a SIP message is traversing a public network. Hiding the address occurs when it is a private address; or when the owner of the private network does not want the IP addresses of their equipment exposed on a public network or on other private networks to which the Net-Net 9200 connects.

The Home Realm's Purpose

A home realm is required because the home address for each SIP NAT is used to create a unique encoding of SIP NAT cookies. You can define the home realm as a network internal to the Net-Net 9200, which eliminates the need for an actual home network connected to the Net-Net 9200. You can define this virtual home network if the supply of IP addresses is limited (because each SIP NAT requires a unique home address), or if all networks to which the Net-Net 9200 is connected must be private to hide addresses.

For example, you can define a public home realm using the loopback network (127.0.0.0) and using the home realm address prefix (for example, 127.0.0.0/8) for encoding addresses that do not match (all addresses outside 127.0.0.0/8) in SIP NAT cookies. The SIP NAT address prefix field can be used to accomplish this while keeping the ability to define an address prefix for the realm for ingress realm determination and admission control. By defining the SIP NAT address prefix as 0.0.0.0, the home realm address prefix is used to encode addresses that do not match.

ACLI Instructions and Examples

To configure the home realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

From this point, you can configure SIP configuration parameters. To view all global SIP configuration parameters, enter a ? at the system prompt.
4. **home-realm-id**—Enter the name of the realm you want to use for the realm ID. For example, **acme**.

The name of the realm must correspond to the identifier value you entered when you configured the realm.
5. **egress-realm-id**—*Optional*. Enter the egress realm ID to define the default route for SIP requests addressed to destinations outside the home realm's address prefix.

If you enter a value for this optional field, it must correspond to the identifier value you entered when you configured the realm.

Note: You should leave this parameter blank for access/backbone applications. When left blank, the realm specified in the home-realm-id parameter is used by default.
6. **nat-mode**—Indicate the SIP NAT mode:
 - **public**—Identifies the subnet defined in the addr-prefix-id field of the home realm is public and the subnet defined in the addr-prefix-id field of all external realms identified in the SIP NAT are private networks. IP addresses are encoded in SIP messages received from the external realm defined by the SIP NAT. The IP addresses are decoded in messages that are sent to the realm.
 - **private**—Identifies the subnet defined in the addr-prefix-id field of the home realm is private and the subnet defined in the addr-prefix-id field of all external realms identified in the SIP NAT are public networks. IP addresses are encoded in SIP messages sent to the external realm defined by the SIP NAT and decoded in messages received from the realm.
 - **none**—No SIP NAT function is necessary.

The following example shows the SIP home realm parameters noted above as they might be configured for a peering network.

```

sip-config
    home-realm-id          acme
    egress-realm-id
    nat-mode               Public

```

Setting Up a SIP Interface

This section explains how to configure a SIP interface. The SIP interface defines the transport addresses (IP address and port) upon which the Net-Net 9200 receives and sends SIP messages.

Overview

The SIP interface defines the signaling interface. You can define a SIP interface for each network or realm to which the Net-Net 9200 is connected. SIP interfaces support both UDP and TCP transport, as well as multiple SIP ports (transport addresses). The SIP interface also lets Hosted NAT Traversal (HNT) be used in any realm.

The SIP interface configuration process involves configuring the following features:

- address and transport protocols (SIP ports)
- redirect action
- proxy mode
- trust mode

About SIP Ports

A SIP port defines the transport address and protocol the Net-Net 9200 uses for a SIP interface for the realm. A SIP interface has one or more SIP ports to define the IP address and port upon which the Net-Net 9200 sends and receives messages. For TCP, it defines the address and port upon which the Net-Net 9200 listens for inbound TCP connections for a specific realm.

You need to define at least one SIP port on which the SIP proxy listens for connections. If you are using both UDP and TCP, you must configure more than one port. For example, if a call is sent to the Net-Net 9200 using TCP, which it needs to send out as UDP, two SIP ports are needed.

Preferred SIP Port

When a SIP interface contains multiple SIP ports of the same transport protocol, a preferred SIP port for each transport protocol is selected for outgoing requests when the specific SIP port cannot be determined. When forwarding a request that matched a cached registration entry (HNT or normal registration caching), the SIP port upon which the original REGISTER message arrived is used. Otherwise, the preferred SIP port for the selected transport protocol is used. When selecting the preferred SIP port, the default SIP port of 5060 will be selected over other non-default ports.

For SIP interfaces using the SIP NAT function, the preferred SIP port address and port will take precedence over the external address of the SIP NAT when they do not match. If both TCP and UDP SIP ports are defined, the address and port of the preferred UDP port is used.

UDP to TCP Failover

With the maximum UDP length option is configured, when the Net-Net SBC forwards a message larger than the value specified in the maximum UDP length

parameter, it attempts to open an outgoing TCP connection to do so. This connection might fail for a number of reasons; for example, an endpoint might not support TCP, or it might be behind a firewall. When the option is not configured, the Net-Net SBC's default behavior is to return the SIP status message **513 Message too Large**.

You can set the global SIP configuration's `max-udp-length` option for global use in the SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.

- `max-udp-length=x` (where X is the maximum length)—Sets the largest UDP packets that the Net-Net SBC will pass. Packets exceeding this length trigger the establishment of an outgoing TCP session to deliver the packet; this margin is defined in RFC 3261. The system default for the maximum UDP packet length is 1500.

Proxy Mode

The Net-Net 9200's proxy mode determines whether it forwards requests received on the SIP interface to target(s) selected from local policy; or sends a redirect response to the previous hop. Sending the redirect response causes the previous hop to contact the targets directly.

If the source of the request matches a session agent with a proxy mode already defined, that mode overrides the proxy mode defined in the SIP interface.

You can configure the proxy mode to use the Record-Route option. Requests for stateless and transaction operation modes are forwarded with a Record-Route header that has the Net-Net 9200's addresses added. As a result, all subsequent requests are routed through the Net-Net 9200.

Redirect Action

The redirect action is the action the SIP proxy takes when it receives a SIP Redirect (3xx) response on the SIP interface. If the target of the request is a session agent with redirect action defined, its redirect action overrides the SIP interface's.

You can set the Net-Net 9200 to perform a global redirect action in response to Redirect messages.

The default behavior of the Net-Net 9200 is to recurse on SIP Redirect responses received from the user agent server (UAS) and send a new request to the Contact headers contained in the SIP Redirect response.

Instead of this default behavior, the Net-Net 9200 can proxy the SIP Redirect response back to the user agent client (UAC) using the value in the session agent's `redirect action` field (when the UAS is a session agent). If there are too many UASes to define as individual session agents or if the UASs are HNT endpoints, and SIP Redirect responses need to be proxied for UASs that are not session agents; you can set the default behavior at the SIP Interface level.

SIP maddr Resolution

The Net-Net SBC provides enhanced resolution of addresses found in SIP contact headers, or in the `maddr` (multicast address) parameter of SIP 3xx REDIRECT messages. Previous releases resolved these addresses as either a host address or as a session agent name. With this release these addresses can also be resolved as session agent group (SAG) names.

Support for SAG-based resolution is provide by a new **sip-config** parameter, **sag-lookup-on-redirect**. By default, SAG lookup is disabled, providing compatibility with prior releases.

The following sample SIP REDIRECT and ACLI configuration fragment illustrate enhanced processing.

RTN 1798

Status-Line: SIP/2.0 302 Moved

Message Header

```
Via: SIP/2.0/UDP
192.168.200.224:5060;branch=z9hG4bKa0fs40009o90sc8oo780.1
From: <sip:1111@192.168.1.222:6000>;tag=1
To: sut <sip:2223@192.168.1.224:5060>;tag=11
Call-ID: 1-28515@192.168.1.222
CSeq: 1 INVITE
Contact: <sip:1111@192.168.1.223;maddr=test.acmepacket.com>
Privacy: user,id;critical;session
P-Preferred-Identity: sipp <sip:sipp@192.168.200.222:5060>
P-Asserted-Identity: abc.com
Subject: abc
Proxy-Require: privacy,prack,abc
Content-Length: 0
```

session-group

group-name	test.acmepacket.com
description	
state	enabled
app-protocol	SIP
strategy	Hunt
dest	
	192.168.200.222
	192.168.200.223

...

...

In this case, when the SBC receives the 302, it resolves the information from *maddr* to a SAG name. In the above example, it will resolve to the configured SAG – *test.acmepacket.com*. The destinations configured in SAG *test.acmepacket.com* will be used to route the call.

SAG-based address resolution is based on the following set of processing rules.

1. When the Contact URI does not have an *maddr* parameter, and the hostname is not an IP Address, the Net Net SBC will look for a SAG matching the hostname.
2. When the Contact URI has an *maddr* parameter that contains an IP address, the Net Net SBC will not look for a SAG; it will use the IP Address as the target/next-hop.
3. When the Contact URI has an *maddr* parameter that contains a non-IP-address value, the Net Net SBC will look for a SAG matching the *maddr* parameter value.

The above logic can be turned on by enabling *sag-lookup-on-redirect* in the *sip-config* object as shown below.

ACLI Configuration and Examples

To configure the Net-Net SBC to perform SAG-based *maddr* resolution:

1. From superuser mode, use the following command sequence to access *sip-config* configuration mode. While in this mode, you configure SAG-based address resolution.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

2. Use the **sag-lookup-on-redirect** parameter to enable SAG-based *maddr* resolution.

```
ACMEPACKET(sip-config)# sag-lookup-on-redirect enabled
ACMEPACKET(sip-config)#
```

Use **done**, **exit**, and **verify-config** to complete SAG-based address resolution.

Trust Mode

The Net-Net 9200 supports the Calling Identity privacy requirements based on RFC 3323 and RFC 3325. The trust mode in the SIP interface determines whether the source and destination of a request are trusted entities. With the implementation of this feature, the Net-Net 9200 can understand and support the privacy headers and provide the capability for anonymous packets.

The Net-Net 9200, which acts as a boundary device between the trusted platform and the untrusted Internet, understands the following headers:

- Privacy
- P-Asserted-Identity
- P-Preferred-Identity

Depending on the value of these headers and the mode in which the Net-Net 9200 is being operated (B2BUA or proxy), the appropriate actions are performed.

Note: Please see [P-Asserted-ID Output to CDRs \(526\)](#) for information on writing P-Asserted-ID headers to CDRs.

About the Process

On receiving a message, the Net-Net 9200 checks whether the message source is trusted or not. It checks the SIP interface's trust mode value and, if the source is a session agent, the session agent's trust me value. Depending on these values, the Net-Net 9200 decides whether the request's or response's source is trusted. If it receives message from a trusted source and the message contains the P-Asserted-Identity header field, the Net-Net 9200 passes this message to the outgoing side. The outgoing side then decides what needs to be done with this request or response.

If the request or the response is received from an untrusted source, the Privacy header value is *id* (privacy is requested), and the P-Asserted-Identity header field is included, the Net-Net 9200 strips the Privacy and the P-Asserted-Identity headers and passes the request or the response to the outgoing side.

If the request or the response contains the P-Preferred-Identity header and the message source is untrusted, the Net-Net 9200 strips the P-Preferred-Identity header from the request or the response and passes the message to the outgoing side.

If the source is trusted or privacy is not requested (the value of the Privacy Header is not *id*) and the request or the response contains the P-Preferred-Identity header, the Net-Net 9200 performs the following actions:

- inserts the P-Asserted-Identity header field with the value taken from the P-Preferred-Identity header field
- deletes the P-Preferred-Identity header value
- passes this request or the response to the Outgoing side for the appropriate action, depending on the whether the destination is trusted or not

After the Net-Net 9200 passes the request or the response to the outgoing side, it checks whether the destination is trusted by checking the SIP interface's trust mode value and the session agent's trust me value (if the destination is configured as session agent).

- The destination is trusted

The Net-Net 9200 does nothing with the request or the response and passes it to the destination. If the P-Asserted-Identity headers are present, they are passed to the session agent (if the destination is configured as session agent).

- The destination is untrusted

The Net-Net 9200 looks at the value of the Privacy header. If set to *id*, the Net-Net 9200 removes all the P-Asserted-Identity headers (if present). It strips the Proxy-Require header if it is set to *privacy*. The Net-Net 9200 also sets the From field of SIP header to *Anonymous* and strips the Privacy header.

If the Privacy header is set to *none*, the Net-Net 9200 does not remove the P-Asserted-Identity header fields.

If there is no Privacy header field, the Net-Net 9200 will not remove the P-Asserted-Identity headers.

To implement this feature, you need to configure the session agent's **trust-me** parameter to enabled (if the message source is a session agent) and the SIP interface's **trust-mode** to the appropriate value.

ACLI Instructions and Examples

To configure a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

From this point, you can configure SIP interface parameters.
4. **state**—Set the SIP interface to enabled or disabled.
5. **realm-id**—Enter the name of the realm to which the SIP interface is connected.
6. **description**—Enter an optional description for this SIP interface. If your entry contains any spaces, you must put the entire entry in double quotation marks.

7. **sip-ports**—See the following section for instructions.
8. **carriers**—Enter the list of carriers related to the SIP interface.

Entries in this field can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! " \$ % ^ & * () + - = < > ? ' | { } [] @ / \ ' ~ , . _ ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carrier are valid carrier field formats.

You can use the **add** and **delete** commands when you enter the **carriers** parameter to edit the carriers list; entering a new list of values without the **add** or **delete** commands overwrites the entire list.

ACMEPACKET(sip-interface)# **carriers add acme_carrier**

9. **proxy-mode**—Enter a option for the proxy mode parameter.

Options	Description
Proxy	Forward all SIP requests to selected targets.
Redirect	Send a SIP 3xx redirect response with the selected target(s) in the Contact header.
Record-Route	Forward requests to selected target(s) and insert a Record-Route header with the Net-Net 9200's address. For stateless and transaction mode only.

10. **redirect-action**—Enter the value for the redirect action.

Options	Description
Proxy	Send the SIP request back to the previous hop.
Recurse	Recurses on the Contacts in the response.

The designated proxy action applies to SIP 3xx responses received from non-session agents and to 3xx responses received from session agents without configured SIP Redirect message actions (for example, session agents without values for the redirect action field).

11. **contact-mode**—Enter the Contact header routing mode, which determines how the contact address from a private network is formatted.

For example, whether a maddr parameter equal to the Net-Net system's SIP interface needs to be added to a URI present in a Contact header.

The available values are:

Options	Description
none	The address portion of the header becomes the public address of that private realm.
maddr	The address portion of the header will be set to the IP address of the Net-Net system's B2BUA.
strict-route	A Route header is not included in the outgoing request.
loose-route	The Record-Route header is included in a Request, which means the destination of the request is separated from the set of proxies that need to be visited along the way.

12. **nat-traversal**—Define the type of HNT enabled for SIP. Available values are:

Options	Description
none	HNT function disabled for SIP (default value).
rport	SIP HNT function only applies to endpoints that include the rport parameter in the Via header. HNT applies when the source IP address of the SIP request does not match the IP address in the Via header.
always	SIP HNT applies to requests with a source IP address that does not match the IP address in the top Via header (even when the rport parameter is not present) and the value in the sent-by of the top Via header matches the host port value in the Contact URI.

13. **nat-interval**—Set the expiration time in seconds for the Net-Net 9200's cached registration entry for an HNT endpoint. Available values include:

- minimum: zero (0) seconds (default value)
- maximum: 999999999 seconds

Acme Packet recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to cause the UA to send REGISTER messages frequently enough to retain the port binding in the NAT. Retaining the binding lets inbound requests to be sent through the NAT.

14. **registration-caching**—Enable for use with all UAs, not just those that are behind NATs. By default, this field is set to disabled. If enabled, the Net-Net 9200 caches the Contact header in the UA's REGISTER request when it is addressed to one of the following:

- Net-Net 9200
- registrar domain value
- registrar host value

The Net-Net 9200 then generates a Contact header with the Net-Net 9200's address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.

Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:

- HNT enabled: the Net-Net 9200 takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.
- HNT disabled: the user part of the Contact header is taken from the URI in the From header and no cookie is appended. This is the default behavior of the Net-Net 9200.

When the registrar receives a request that matches the Address of Record (AoR, the To header in the REGISTER message), it sends the matching request to the Net-Net 9200, which is the Contact address. Then, the Net-Net 9200 forwards the request to the Contact-URI it cached from the original REGISTER message.

15. **min-reg-expire**—Set the time in seconds for the minimum registration expiration time for HNT registration caching.

- minimum: zero (0) seconds
- maximum: 999999999 seconds

The default value is 300.

This value defines the minimum expiration value the Net-Net 9200 places in each REGISTER message it sends to the real registrar. In HNT, the Net-Net 9200 caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the Net-Net 9200 to send frequent registrations on to the real registrar.

16. **registration-interval**—Set the Net-Net 9200's cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the Net-Net 9200 to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires.

- minimum: zero (0) seconds

A registration interval of zero causes the Net-Net 9200 to pass back the expiration time set by and returned in the registration response from the registrar.

- maximum: 999999999 seconds

The default value is 3600 seconds (one hour). If the expiration time you set is less than the expiration time set by and returned from the real registrar, the Net-Net 9200 responds to the refresh request directly rather than forwarding it to the registrar.

Although the registration interval applies to non-HNT registration cache entries, and the loosely related NAT interval applies to HNT registration cache entries, you can use the two in combination. Using a combination of the two means you can implement HNT and non-HNT architectures on the same Net-Net 9200. You can then define a longer interval time in the registration interval field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

17. **route-to-registrar**—Enable routing to the registrar to send all requests that match a cached registration to the destination defined for the registrar host; used when the Request-URI matches the registrar host value or the registrar domain value, not the Net-Net 9200's address. Because the registrar host is the real registrar, it should send the requests back to the Net-Net 9200 with the Net-Net 9200's address in the Request-URI. By default, this field is set to `disabled`.

For example, you should enable routing to the registrar if your network uses a Net-Net 9200 and needs requests to go through its service proxy, which is defined in the registrar host field.

18. **uri-fqdn-domain**—Set to enabled to use FQDNs in SIP URIs.

If set to enabled, and used with an FQDN domain/host, the requests generated by the Net-Net 9200 on this SIP interface will have the host part of the URI set to this FQDN value. Only the Request, To, and From URIs are changed. After the dialog is established, the URIs are not changed.

19. **options**—*Optional*.

20. **trust-mode**—Set the trust mode for the SIP interface, which is checked by the Net-Net 9200 when it receives a message to determine whether the message source is trusted. Available options are:

Options	Description
all	Trust all SIP elements (sources and destinations) in the realm(s), except untrusted session agents. Untrusted session agents are those that have the <code>trust me</code> parameter set to <code>disabled</code> .
agents-only	Trust only trusted session agents. Trusted session agents are those that have the <code>trust me</code> parameter set to <code>enabled</code> .
realm-prefix	Trust only trusted session agents, and source and destination IP addresses that match the SIP interface's realm (or subrealm) address prefix. Only realms with non-zero address prefixes are considered.
registered	Trust only trusted session agents and registered endpoints. Registered endpoints are those with an entry in the Net-Net 9200's registration cache.

Session agents must have one or more of the following:

- global realm
- same realm as the SIP interface
- realm that is a subrealm of the SIP interface's realm

Configuring SIP Ports

To configure SIP ports:

1. From sip-interface, type **sip-ports** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-interface)# sip-ports
ACMEPACKET(sip-port)#
```

2. **address**—Enter the IP address on which to listen. For example:
192.168.11.101
3. **port**—Enter the port number you want to use. The available values include:
 - minimum: 1025
 - maximum: 65535

The default value is 5060.
4. **transport-protocol**—Indicate the transport protocol you want to associate with the SIP port. The available values include:
 - TCP: provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with the retransmission of packets when necessary.
 - UDP: provides a simple message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
5. **allow-anonymous**—Define the allow anonymous criteria for accepting and processing a SIP request from another SIP element. If a request cannot be accepted because of the criteria set here, it is rejected with a 403 (Forbidden) response.

The anonymous connection mode criteria includes admission control based on whether an endpoint has successfully registered. Requests from an existing SIP dialog are always accepted and processed. Available options are:

Options	Description
all	All requests from any SIP element are allowed.
agents-only	Only requests from configured session agents are allowed. The session agent must fit one of the following criteria: • have a global realm • have the same realm as the SIP interface • be a sub-realm of the SIP interface's realm.
realm-prefix	The source IP address of the request must fall within the realm's address prefix or a SIP interface sub-realm. A sub-realm is a realm that falls within a <i>realm group</i> tree. The sub-realm is a child (or grandchild, and so on) of the SIP interface realm. Only realms with non-zero address prefixes are considered. Requests from configured session agents (as described in the agents-only option) are also allowed.

Options	Description
registered	Only requests from user agents that have an entry in the registration cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed from any user agent. The registration cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the agents-only option) are also allowed.
register-prefix	Only requests from user agents that have an entry in the Registration Cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed only when the source IP address of the request falls within the realm address-prefix or a SIP Interface sub-realm. Only realms with non-zero address prefixes are considered. The Registration Cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the agents-only option) are also allowed.

SIP Timers

This section explains how to configure SIP retransmission and expiration timers.

Overview

SIP timers define the transaction expiration timers, retransmission intervals when UDP is used as a transport, and the lifetime of dynamic TCP connections. The retransmission and expiration timers correspond to the timers defined in RFC 3261.

- **init-timer**—Initial request retransmission interval. It corresponds to Timer T1 in RFC 3261.

This timer is used when sending requests over UDP. If the response is not received within this interval, the request is retransmitted. The retransmission interval is doubled after each retransmission.

- **max-timer**—Maximum retransmission interval for non-INVITE requests. It corresponds to Timer T2 in RFC 3261.

The retransmission interval is doubled after each retransmission. If the resulting retransmission interval exceeds the max timer, it is set to the max timer value.

- **trans-expire**—Transaction expiration timer. This value is used for timers B, D, F, H and J as defined in RFC 3261.
- **invite-expire**—Defines the transaction expiration time for an INVITE transaction after a provisional response has been received. This corresponds to timer C in RFC 3261.

If a final response is not received within this time, the INVITE is cancelled. In accordance with RFC 3261, the timer is reset to the invite expire value when any additional provisional responses are received.

- **inactive-dynamic-conn-timer**—Defines the idle time (no active sessions and no SIP messages sent or received) of a dynamic TCP connection before the connection is torn down. There is no timer in RFC 3261 corresponding to this function.

ACLI Instructions and Examples

To configure timers:

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

ACMEPACKET# **configure terminal**

2. Type **session-router** and press <Enter>.

ACMEPACKET(configure)# **session-router**

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

ACMEPACKET(session-router)# **sip-config**

ACMEPACKET(sip-config)#

4. **init-timer**—Enter the initial timeout (T1) value for SIP in milliseconds. This value is applied to responses to INVITE requests and any UDP SIP requests. The available values include:

- minimum: 0 milliseconds
- maximum: 999999999 milliseconds

The default value is 500 milliseconds.

5. **max-timer**—Enter the maximum transmission timeout (T2) for SIP in milliseconds.

When sending SIP over UDP, a re-transmission timer is used. If the timer expires and the message is re-transmitted, the re-transmission timer is then set to twice the previous value (but will not exceed the maximum timer value). Using the default values of 500 milliseconds and 4000 milliseconds, the re-transmission timer is 0.5, then 1, 2, and finally 4. The incrementing continues until the transmission expire timer activates.

- minimum: 0 milliseconds
- maximum: 999999999 milliseconds

The default value is 4000 milliseconds.

6. **trans-expire**—Enter the transaction expire timeout value (Timer B) in seconds to set the time for SIP transactions to live. The same value is used for Timers D, F, H and J.

- minimum: 0 seconds
- maximum: 999999999 seconds

The default value is 32 seconds.

7. **invite-expire**—Enter the invite expire timeout value (Timer C) in seconds to indicate the time for SIP client transaction will live after receiving a provisional response.

- minimum: 0 seconds
- maximum: 999999999 seconds

The default value is 180 seconds.

8. **inactive-dynamic-conn**—Enter the inactive dynamic connection value in seconds to set the time limit for inactive dynamic connections.

If the connection between the SIP proxy and a session agent is dynamic (for example, through dTCP), and the connection has been idle for the amount of time specified here, the SIP proxy breaks the connection.

- minimum: 1 second
- maximum: 999999999 seconds

The default value is 32 seconds.

The following example shows the above SIP configuration timer values for a peering network. Some parameters are omitted for brevity.

```

sip-config
    init-timer                500
    max-timer                 4000
    trans-expire              32
    invite-expire             180
    inactive-dynamic-conn     32

```

Configuring Timers for a sip-interface

You can set the expiration times for SIP messages, and you can set a counter that restricts the number of contacts that the Net-Net SBC tries when it receives a 3xx-class SIP response.

If you are familiar with the ACLI **trans-expire** and **invite-expire** timers in the global SIP configuration, the two timer parameters will look familiar and operate in a similar way.

ACLI Instructions and Examples

To set timers and counters for SIP interfaces:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the signaling-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-interface)#
```

If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI **select** command) the SIP interface that you want to edit.

4. **trans-expire**—Enter the value in seconds for SIP transactions. This timer controls the following timers specified in RFC 3261:
 - Timer B—SIP INVITE transaction timeout
 - Timer F—non-INVITE transaction timeout
 - Timer H—Wait time for ACK receipt
 - Timer TEE—Used to transmit final responses before receiving an ACK

You can leave this parameter set to 0 (default), or enter a value between 0 and 999999999. If you set this parameter to 0, then the Net-Net SBC uses the timer value from the global SIP configuration.

5. **invite-expire**—Enter the value in seconds for a SIP client/server transaction after receiving a provisional response.

You set this timer for the client and the sever by configuring it on the SIP interface that corresponds to a realm.

You can leave this parameter set to 0 (default), or enter a value between 0 and 1000000000. If you set this parameter to 0, then the Net-Net SBC uses the timer value from the global SIP configuration.

6. **max-redirect-contacts**—Enter the maximum number of contacts or routes for the Net-Net SBC to attempt in when it receives a SIP redirect (3xx response). When this number is exceeded, the Net-Net SBC sends a “408 Exhausted Redirects” error to the endpoint. If you leave this parameter set to 0 (default), then the Net-Net SBC will exercise no restrictions on the number of contacts or routes. The value range for this parameter is 0 to 10.
7. Save your work using the CLI **done** command.
8. Save and activate your configuration.

SIP Options Tag Handling

This section explains how to configure SIP options on a global or per-realm level and how to specify whether the feature treatment applies to traffic inbound or to outbound from a realm, or both.

SIP extensions that require specific behavior by UAs or proxies are identified by option tags. Option tags are unique identifiers used to designate new options (for example, extensions) in SIP. These option tags appear in the Require, Proxy-Require, and Supported headers of SIP messages.

Option tags are compatibility mechanisms for extensions and are used in header fields such as Require, Supported, Proxy-Require, and Unsupported in support of SIP.

The option tag itself is a string that is associated with a particular SIP option (i.e., an extension). It identifies this option to SIP endpoints.

Overview

The SIP specification (RFC 3261) requires that the Net-Net 9200 B2BUA reject any request that contains a Require header with an option tag the Net-Net system does not support. However, many of these extensions operate transparently through the Net-Net system's B2BUA. You can configure how SIP defines the Net-Net system's B2BUA treatment of specific option tags.

Also, there might be certain extensions that an endpoint indicates support for by including the option tag in a Supported header. If you do not want a given extension used in your network, then you can configure SIP option tag handling to remove the undesired option tag from the Supported header. You can also specify how option tags in Proxy-Require headers are to be treated.

Configuration Overview

You configure the SIP feature element to define option tag names and their treatment by the Net-Net system when the option tag appears in a Supported header, a Require header, and a Proxy-Require header. If an option tag is encountered that is not configured as a SIP feature, the default treatments apply. You only need to configure option tag handling in the SIP feature element when non-default treatment is required.

You can specify whether a SIP feature should be applied to a specific realm or globally across realms. You can also specify the treatment for an option based on whether it appears in an inbound or outbound packet. Inbound packets are those that are coming from a realm to the Net-Net 9200 and outbound packets are those which are going from the Net-Net 9200 to the realm.

ACLI Instructions and Examples

To configure SIP option tag handling:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-feature** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-feature
```

```
ACMEPACKET(sip-feature)#
```

From this point, you can configure SIP option tags parameters. To view all sip-feature parameters, enter a ? at the system prompt.

4. **name**—enter a name for the option tag that will appear in the Require, Supported, or Proxy-Require headers of inbound and outbound SIP messages.

You must enter a unique value.

Note: Valid option tags are registered with the IANA Protocol Number Assignment Services under Session Initiation Protocol Parameters. Because option tags are not registered until the SIP extension is published as a RFC, there might be implementations based on Internet-Drafts or proprietary implementations that use unregistered option tags.

5. **realm**—Enter the name of the realm with which this option tag will be associated. If you want to apply it globally, leave this parameter blank.
6. **support-mode-inbound**—*Optional*. Set the support mode that defines how the option tag is treated when encountered in an inbound SIP message's Supported header. Available values include:
 - **pass** (default)—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **strip**—Indicates the tag should not be included in the outgoing message. Use strip if you do not want the extension used.
7. **require-mode-inbound**—*Optional*. Set the require mode that defines how the option tag is treated when it is encountered in an inbound SIP message's Require header.
 - **pass**—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **reject** (default)—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.
8. **require-mode-inbound**—*Optional*. Set the require proxy mode that defines how the option tag is treated when encountered in an incoming SIP message's Proxy-Require header.
 - **pass** (default)—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **reject**—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

9. **support-mode-outbound**—*Optional*. Set the support mode that defines how the option tag is treated when encountered in an outbound SIP message's Supported header. Available values include:
 - `pass` (default)—Indicates the B2BUA should include the tag.
 - `strip`—Indicates the tag should not be included in the outgoing message. Use `strip` if you do not want the extension used.
10. **require-mode-outbound**—*Optional*. Set the require mode that defines how the option tag is treated when it is encountered in an outbound SIP message's Require header.
 - `pass`—Indicates the B2BUA should include the tag.
 - `reject` (default)—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.
11. **require-mode-outbound**—*Optional*. Set the require proxy mode that defines how the option tag is treated when encountered in an outgoing SIP message's Proxy-Require header.
 - `pass` (default)—Indicates the B2BUA should include the tag.
 - `reject`—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

The following example shows SIP option tag handling configured for non-default treatment of option tags.

```
sip-feature
    name                      newfeature
    realm                     peer-1
    support-mode-inbound      Strip
    require-mode-inbound      Reject
    proxy-require-mode-inbound Pass
    support-mode-outbound     Pass
    require-mode-outbound     Reject
    proxy-require-mode-outbound Reject
```

SIP Options

This section explains how you can configure a limited list of specialized SIP features and/or parameters called options. The options described here were developed to meet specific needs not addressed by the standard SIP configuration parameters. Not all users have a need for these options.

Note: Acme Packet recommends checking with your Acme Packet representative before applying any of these options.

Overview

You can configure options for the SIP configuration and SIP interface. Both elements include a parameter (**options**) that you use to configure the options described in the following section.

Global SIP Options

The following table lists the SIP options supported by the Net-Net 9200.

Option	Description
add-error-to-tag=no	If present (even when set to no), suppresses the addition of an Acme tag on 3xx-6xx responses.
add-prov-to-tag=no	Prevents the Net-Net 9200 from adding a tag parameter to the To header (to-tag) to non-100 provisional responses to INVITE requests. Used when a provisional (101-199) response is received from the UAS on a client transaction without a to-tag. By default, the Net-Net 9200 adds the tag cookie in the response (as though it had a tag) sent back to the UAC for the associated server transaction. When you include this option in the SIP configuration, and the response from the UAS does not have a to-tag, the response forwarded to the UAC will not have a to-tag.
add-reg-expires	Causes an Expires header to always be included in a REGISTER response with the registration caching and HNT traversal functions of the Net-Net 9200. Use for endpoints that do not understand the expires parameter in the Contact header.
add-ruri-user=<methods>	Causes a userinfo portion to be added to a Request-URI when one is not present. Used to support the OKI phone, which registers a Contact of just an IP-Address but rejects initial INVITEs if the Request_URI does not have a userinfo part. <methods> is a comma-separated list of methods to which the option should apply. If more than one method is listed, the list must be enclosed in quotes. This option only applies to out-of-dialog requests (no tag parameter in the To header). However, if ACK is listed, it will apply to all ACK requests because an ACK is always supposed to have a to-tag.
allow-notify-no-contact	Prevents the Net-Net 9200 from rejecting illegally constructed NOTIFYs with a 400 Bad Request response. NOTIFY requests without Contact header are allowed to pass through the Net-Net 9200 instead.

Option	Description
call-id-host=<host>	Causes the Net-Net 9200 to include a host part (ID@host) in the Call-ID it generated. <host> is the hostname (or IP address) that is to appear in the host part of the Call-ID. If not specified, the SIP port address is used.
disable-privacy	Prevents the change of the P-Preferred-Identity to P-Asserted-Identity and lets the P-Preferred-Identity go through unchanged.
drain-sendonly	Causes the Net-Net 9200 to examine the SDP attributes and change sendonly mode to sendrecv. This causes the endpoint receiving the SDP to send RTP, which is required for HNT traversal endpoints to work with media servers. The Net-Net 9200 sets up the flow so that RTP coming from the endpoint are dropped to prevent the UA that sent the sendonly SDP from receiving packets. See the option video-sbc-session also.
encode-contact=<prefix>	Causes the Net-Net 9200 to encode Contact addresses into the userinfo part of the URI. It applies only to Contact address that usually get the maddr parameter. Use when the Net-Net 9200 needs requests sent to the URI in the Contact sent instead to the Net-Net 9200. The host part of the URI will have the Net-Net 9200's address. The <prefix> serves as a place between the original userinfo and the encoded address. If a <prefix> is specified, a default of +9200 is used. Without this option, the Net-Net 9200 adds a maddr parameter.
fix-to-header	For requests that have the Net-Net- 9200's address in both the Request-URI and the To-URI, it sets the host port of the To-URI to a local policy's next hop target on out-of-dialog requests (no to-tag).
forward-reg-callid-change	Addresses the case when an endpoint reboots and performs a third party registration before its old registration expires. During this reregistration, the contact header is the same as it was pre-reregistration. As a consequence of the reboot, the SIP Call-ID changes. In this situation, the Net-Net 9200 does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI. To remedy this problem, the Net-Net 9200 now keeps track of the Call-ID in its registration cache. A new option in the sip-interface configuration element forces the Net-Net 9200 to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.
global-contact	Addresses interoperability in the Dialog and Presence event packages that are used in hosted PBX and IP centrex offerings. This option enables persistent URIs in the Contact headers inserted into outgoing SIP messages. If this option is not used, URIs placed in the Contact header of outgoing messages are only valid within the context of the dialog to which the message is associated.
load-limit=<cpu-percent>	Defines the CPU usage percentage at which the Net-Net 9200 should start rejecting calls. Default value is 95%.

Option	Description
gateway-contact=<contact ID> (session agent)	Defines the URI parameter into which the gateway's address is encoded.
load-limit=<cpu-percent>	Defines the CPU usage percentage at which the Net-Net 9200 should start rejecting calls. Default value is 95%.
gateway-contact=<contact ID> (session agent)	Defines the URI parameter into which the gateway's address is encoded.
max-register-forward=<value>	Defines a limit (as assigned in the value field) of REGISTERS to be forwarded to the registrar. During each second, the sipd counts how many REGISTERS have been sent to the registrar. It checks the threshold when it receives a REGISTER from the UA and determines that less than half the real registration lifetime is left. If the number of REGISTERS forwarded (new and updates) in the current second exceeds the configured threshold, it will respond to the UA from the cache.
max-register-refresh=<value>	Defines the desired limit of REGISTER refreshes from all the UAs. Each second of time, sipd counts the number of REGISTER/200-OK responses sent back. When the threshold is exceeded, it increments the expire time (based on NAT interval) by one second and resets the count. By default no threshold is applied. The recommended value is somewhat dependent on the Net-Net 9200 hardware used, but 300 can be used as an initial value.
max-routes=<number of routes>	Restricts the number of routes through which the sipd will iterate from a local policy lookup. For example, setting this option to 1 causes the Net-Net 9200 to only try the first, best, route. Setting this option to 0, or omitting it, lets the Net-Net 9200 use all of the routes available to it (with the priority scheme for route matching). When you test a policy using the test-policy CLI command, this option is not recognized and all options that match the criteria are displayed.
max-udp-length=<bytes>	Lets you override the default value of the maximum UDP datagram size (1500 bytes; sipd requires the use of SIP/TCP at 1300 bytes). Setting this option to zero (0) forces sipd to send fragmented UDP packets.

Option	Description
<code>media-release="<header-name>[;<header-param>]"</code>	Enables the multi-system media release feature that encodes IP address and port information for the media streams described by SDP. It lets another Net-Net 9200 decode the data to restore the original SDP to allow the media to flow directly between endpoints, which is being serviced by multiple Net-Net SDs. The media release information can appear in the following places: • SIP header P-Media-Release: <encoded-media-interface-information> • Header parameter on a SIP header Contact: <sip:1234@abc.com> ; acme-media=<encoded-media-interface-information> • SDP attribute in the message body a=acme-media: <encoded-media-interface-information> Option includes the following: • <header-name> is SIP header in which to put the information or the special value sdp, which indicates the information should be put into the SDP. • <header-param> is the header parameter name in which to put the information or in the case of the special header name value sdp, it is the SDP attribute name in which to put the information. They identify to where the encoded information is passed. If you do not specify a header, P-Media-Release is used.
<code>reg-cache-mode=<mode></code>	Effects how the userinfo part of Contact address is constructed with registration caching. <mode> values are: • none: userinfo from the received (post NAT) Contact is retained • from: userinfo from the From header is copied to the userinfo of the forwarded Contact header • append: append the UA's Contact address into a cookie appended to the userinfo from the original Contact userinfo. For HNT, the NAT/firewall address is used. • append-from: takes userinfo from the From header and appends the encrypted address to the userinfo from the original Contact userinfo. For HNT, the NAT/firewall address is used. The from mode is used with softswitches that do not use the cookies used by the Net-Net 9200. It also helps limit the number of bytes in the userinfo; which might create duplicate contacts. For example, if the Net-Net 9200's address is 1.2.3.4, both 1234@5.6.7.8 and 1234@4.3.2.1 will result in a Net-Net 9200 contact of 1234@5.6.7.8.
<code>register-grace-timer</code>	Makes the grace time for the SIP Registration configurable. You can configure the grace timer in seconds.
<code>reinvite-trying=<yes></code>	Causes the Net-Net 9200 to send a 100 Trying for re-INVITES, which is normally suppressed. If you enter the option name but omit the value yes, the option is still active.

Option	Description
reject-interval=<value>	When rejecting calls because of CPU load limiting, the Net-Net 9200 adds a "Retry-After" parameter to the error response (typically 503 Service Unavailable). By default the Net-Net 9200 sets the Retry-After value to be 1/10th of the current rejection rate. The value you enter for this option is a multiplier used for increasing the value presented to the UAC in the Retry-After field. For example, if reject-interval=5 (reject interval is set to 10); at a 90% rejection rate the Net-Net 9200 sends "Retry-After: 45".
reject-register=no	Allows REGISTER messages through even during load limiting. By default, REGISTER messages are subject to load limiting.
response-for-not-found	Change the 404 Not Found generated by the Net-Net 9200 to a different response code.
route-register-no-service-route	Prevents the use of the Service-Route procedure to route the subsequent Register requests. If this option is not configured, the Net-Net 9200 routes the Register requests using the Service-Route procedure.
set-inv-exp-at-100-resp	Set Timer C when a 100 Trying response is received (instead of waiting until 1xx (> 100) is received). If the Net-Net 9200 does not receive a 100 Trying response within Timer B then the call should be dropped because there is a problem communicating with the next hop.
strip-domain-suffix-route	Causes sipd to strip any Route headers from the inbound messages coming to the external address of a SIP NAT; if the message contains a FQDN that matches the configured domain suffix for that SIP NAT.
video-sbc-session	Use with drain-sendonly. Provides conference floor support. When configured with drain-sendonly and when the Net-Net 9200 receives an SDP, the Net-Net 9200 proxies the m=control and its related a= and c= unchanged. Although media streams are allocated for this m line, an actual flow is not set up. SDP received with the following: m=video a=sendonly is sent out as the following: m=video a=sendonly a=X-SBC-Session

SIP Interface Options

The following table lists the SIP interface options supported by the Net-Net 9200.

Option	Description
contact-endpoint=<param-name>	Defines a URL parameter to report the real Contact address of an endpoint in a REGISTER message forwarded to a registrar; when the Net-Net is caching registration. (plain or HNT). If <param-name> is not specified, the default value "endpoint" is used. This parameter is added as a URL parameter in the Contact on the REGISTER message. In order for the registration cache to work properly, the softswitch/registrar is expected to include the endpoint parameter in the Request-URI of a SIP request it forwards to the address-of-record.
contact-firewall=<param-name>	Defines a URL parameter to report the NAT between the Net-Net 9200 and the real Contact address of an endpoint in a REGISTRAR message forwarded to a registrar when the Net-Net 9200 is doing registration caching for NHT. If <param-name> is not specified, the default value "firewall" is used. This parameter will be added as a URL parameter in the Contact on the REGISTER message. In order for the registration cache to work properly, the softswitch/registrar is expected to include the endpoint parameter in the Request-URI of any SIP request it forwards for the address-of-record.
contact-vlan=<param-name>	Adds a parameter to outbound Contact-URIs that indicates the VLAN on which the message was received. Use when two networks have overlapping IP space, so contact-firewall and contact-endpoint might have collisions.
response-for-not-found	Change the 404 Not Found generated by the Net-Net 9200 to a different response code.
via-header-transparency	Enables the Net-Net 9200 to insert its Via header on top of the top-most Via header received from user equipment (UE). It then forwards it on to the IP Multimedia Subsystem (IMS) core with the original Via header now located as the bottom-most Via header. The Net-Net 9200 still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.
reg-via-key	Enables non-anonymous INVITES to be referenced in the registration cache from their existing Contact and or Via header.

SIP Session Agent Options

The following table lists the SIP session agent options supported by the Net-Net 9200.

Option	Description
<code>trans-timeouts=<value></code>	Defines the number of consecutive non-ping transaction timeouts that will cause a session agent to be put out of service. The default is 5 (the existing behavior). A value of 0 prevent the session agent from being put out of service due to a non-ping transaction timeout.
<code>via-origin=<parameter-name></code>	Causes a parameter to be included in the top Via header of requests sent to the session agent. The parameter indicates the source IP address of the corresponding request received by the Net-Net 9200. <parameter-name> defines the name of the parameter. If not specified, the default value origin is used.
<code>sip-connect-pbx-reg</code>	Enables SIPConnect feature.

SIP Realm Options

The following table lists the SIP session agent options supported by the Net-Net 9200.

Option	Description
<code>number-normalization</code>	Applies to the SIP To URI. (Currently the Net-Net 9200 supports number normalization on From and To addresses for both inbound and outbound call legs.) Number normalization includes add, delete, and replace string functions that result in consistent number formats. Number normalization occurs on ingress traffic, prior to the generation of accounting records or local policy lookups. (also applies for SIP calls.)
<code>sip-connect-pbx-reg</code>	Enables SIPConnect feature.

SIP Security

This section provides an overview of Net-Net 9200's security capability. Net-Net 9200 security is designed to provide security for VoIP and other multi-media services. It includes access control, DoS attack, and overload protection, which help secure service and protect the network infrastructure (including the Net-Net 9200). In addition, Net-Net 9200 security allows legitimate users to place calls during attack conditions, protecting the service itself.

Net-Net 9200 security includes the Net-SAFE framework's numerous features and architecture designs. Net-SAFE is a requirements framework for the components required to provide protection for the Session Border Controller (SBC), the service provider's infrastructure equipment (proxies, gateways, call agents, application servers, and so on), and the service itself.

Denial of Service Protection

The Net-Net 9200 Denial of Service (DoS) protection functionality protects softswitches and gateways with overload protection, dynamic and static access control, and trusted device classification and separation at Layers 3-5. The Net-Net 9200 itself is protected from signaling and media overload, but more importantly the feature allows legitimate, trusted devices to continue receiving service even during

an attack. DoS protection prevents the Net-Net 9200 host processor from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Volume-based attack (flood) of valid or invalid call requests, signaling messages, and so on.
- Overload of valid or invalid call requests from legitimate, trusted sources

Levels of DoS protection

The multi-level Net-Net 9200 DoS protection consists of the following strategies:

- Fast path filtering/access control: involves access control for signaling packets destined for the Net-Net 9200 host processor as well as media (RTP) packets. The Net-Net 9200 accomplishes media filtering using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted list and a deny list for each Net-Net 9200 realm (although the actual devices can be dynamically trusted or denied by the Net-Net 9200 based on configuration). You do not have to provision every endpoint/device on the Net-Net 9200, but instead retain the default values.
- Host path protection: includes flow classification, host path policing and unique signaling flow policing. Fast path filtering alone cannot protect the Net-Net 9200 host processor from being overwhelmed by a malicious attack from a trusted source. The host path and individual signaling flows must be policed to ensure that a volume-based attack will not overwhelm the Net-Net 9200's normal call processing; and subsequently not overwhelm systems beyond it. The Net-Net 9200 must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent. At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Net-Net 9200 maintains two host paths, one for each class of traffic (trusted and untrusted), with different policing characteristics to ensure that fully trusted traffic always gets precedence.
- Host-based malicious source detection and isolation – dynamic deny list. Malicious sources can be automatically detected in real-time and denied in the fast path to block them from reaching the host processor.

Configuration Overview

NAT table entries are used to filter out undesired IP addresses (deny list). After the packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager based on the sender's IP address. NAT table entries are used to distinguish signaling packets coming in from different sources for policing purposes.

You can configure deny rules based on the following:

- ingress realm
- source IP address
- transport protocol (TCP/UDP)
- application protocol

You can configure guaranteed minimum bandwidth for trusted and untrusted signaling paths.

You can configure signaling path policing parameters for individual source addresses. Policing parameters include:

- peak data rate in bits per second
- average data rate in bits per second
- maximum burst size

Configuring Security

See the [Security \(691\)](#) chapter in this guide for the configuration details.

The SIP NAT Function

This section explains how to configure the optional SIP NAT function. You can configure the SIP NAT function if you need to translate IP address and UDP/TCP port information. The SIP NAT function also prevents private IP addresses in SIP message URIs from traveling through an untrusted network.

Overview

The Net-Net system is an intermediary device that provides NAT functions between two or more realms. It translates IP addresses between untrusted and trusted networks using NAT. A trusted network is inside the NAT, and a untrusted network is outside the NAT. A NAT also lets a single IP address represent a group of computers.

For SIP, the SIP NAT function on the Net-Net 9200 does the following:

- Routes SIP packets between the Net-Net system's SIP proxy (B2BUA) and external networks (or realms), including the translation of IP address and UDP/TCP port information.
- Prevents private IP addresses in SIP message URIs from traveling through the untrusted network. The SIP NAT either translates the private address to one appropriate for an untrusted address or encrypts the private address into the URI.

Packets arriving on the external address are forwarded to the Net-Net system's SIP proxy with the source address changed to the home address. When the Net-Net system's SIP proxy sends packets to the home address, they are forwarded to the external proxy address (and external proxy port), with the source address changed to the external address.

Note: The SIP config's NAT mode parameter works in conjunction with the SIP NAT function configuration. It identifies the type of realm in which the SIP proxy is located (public or private) and affects whether IP addresses in SIP messages are encoded.

The translation of URIs in the actual SIP message occurs as messages are received and sent from the Net-Net system's SIP proxy. For the messages being sent to the external network, the contents of the SIP message are examined after the translation to determine if the destination needs to be changed from the external proxy address to an address and port indicated by the SIP message. This process takes place so the request is sent to where the Request-URI or the Route header indicates, or so the response is sent to where the Via indicates.

NAT Modes

The specific addresses used in translating URIs in the SIP message depend on whether the Net-Net system is performing NAT functions for a trusted or untrusted network. This condition is determined by the NAT mode value you enter when you configure the SIP config element. The NAT modes are:

- **untrusted**—The SIP proxy is associated with an address for an untrusted network (the address value you entered when you configured the SIP interface's SIP port parameter), and the home address in the SIP NAT is the address of the external realm/network. When the URI contains the external address, it is translated to the SIP NAT's home proxy address (or to the SIP port address if the home proxy address field is empty). When a URI contains the external proxy address, it is translated to the home address.

If the URI contains any other private address (matching the realm's address prefix, identified in the SIP NAT's realm ID), it is encrypted and the address is replaced with the home address value. If the URI contains a user part, a suffix consisting of the user NAT tag and the encrypted address is appended to the user part. For example, with a user NAT tag value of `-private-`, the private URI of `sip@123192.169.200.17:5060` will become the public URI of `sip:123-private-eolmhet2chbl3@172.16.0.15`.

If there is no user part, the host consists of the host NAT tag followed by the encrypted address and the domain suffix. A `maddr` parameter equal to the home address (or received in the case of a Via header) is added to the URI. For example, with a host NAT tag value of `PRIVATE-` and a domain suffix value of `private.com`, the private URI of `sip:192.168.200.17:5060` will become the public URI of `sip:PRIVATE-eolmhet2chbl3.private.com:5060;maddr=172.16.0.15`.

- **trusted**—The SIP proxy is on a trusted network (the address value you entered when you configured the SIP interface's SIP port parameter), and the SIP NAT's external address is the public address of the external realm/network. When the URI contains the home address value, it is translated to the value set for the external proxy address. When the URI contains the SIP proxy's address, it is translated to the external address. If the URI contains any other private address (matching the realm's address prefix, identified in the SIP NAT's realm ID), the private address is encrypted and the address is replaced with the external address.

Note: Do not use the home proxy address value with `private` NAT functioning.

Adding an maddr Parameter to a URI

When you configure a SIP interface, you can configure the contact mode. The contact mode sets the contact header routing mode, which determines how the contact address from a trusted network is formatted. You set the contact mode to add a `maddr` parameter equal to the SIP proxy to the URI in the Contact header. For example, the URI from the prior example (`sip:192.168.200.17:5060`) becomes `sip:123-trusted-eolmhet2chbl3@172.16.0.15;maddr=172.16.0.12`.

Note: For SIP elements that do not support the `maddr` parameter, configure a Contact mode as `none`.

You might require this encryption to cause other SIP elements in the untrusted network to send requests directly to the SIP proxy. Otherwise, the requests are sent to the home address. However, responses sent by the SIP proxy will have the SIP proxy's source address, rather than the home address. Some SIP elements might drop responses that come from a IP address different from the one to which the request is sent.

About Headers

You can specify which SIP headers you want effected by the SIP NAT function. The URIs in these headers are translated and encrypted, the encryption occurs according to the rules of this SIP NAT function.

You can enter header values by using either the full header name or its corresponding abbreviation, if applicable. The following table lists the available headers and their corresponding abbreviations

Header	Abbreviation
Call-ID	i
Contact	m
From	f
Record-Route	none
Route	none
Reply-To	none
Replaces	none
Refer-To	r
To	t
Via	v

SIP sessions are terminated and re-originated as new sessions as they are routed through the Net-Net 9200. Among the actions performed, SIP headers are modified to prevent the transmission of IP address and route information.

Replacing Headers

In the SIP signaling message, any Via headers are stripped out and a new one is constructed with the Net-Net 9200's IP address in the sent-by portion. If a Contact header is present, it is replaced with one that has the Net-Net 9200's IP address. All other headers are subject to NATing based on the following rules:

- The Request-URI is replaced with the next hop's IP or FQDN address.
- All other headers are replaced based on the two SIP NAT function SIP NAT function rules

Mapping FQDNs

The Net-Net 9200 maps FQDNs that appear in the certain headers of incoming SIP messages to the IP address that the Net-Net 9200 inserts in outgoing SIP contact headers. The mapped FQDNs are restored in the SIP headers in messages that are sent back to the originator.

This feature is useful to carriers that use IP addresses in the SIP From address to create trunk groups in a softswitch for routing purposes. When the carrier's peer uses FQDNs, the carrier is forced to create trunk groups for each possible FQDN that it might receive from a given peer. Similarly, this can apply to SIP Contact and P-Asserted-Identity headers.

SIP NAT Function Cookies

Cookies are inserted to hide that information is coming from a realm external to the home realm. They are used when information needs to be placed into a given element of a SIP message that must also be seen in subsequent SIP messages within a flow. When forwarding a SIP message, the Net-Net 9200 encodes various information in the outgoing message, which is passed from one side to another in SIP transactions.

SIP NAT function cookies let the Net-Net 9200 hide headers, IP addresses, and SIP URIs. These cookies are included when certain conditions are present in Net-Net system SIP transactions.

Acme Packet's SIP NAT function cookies can be used in the `userinfo`, `host`, `URL` parameter, and `tel` URL parameter portions of the SIP message.

userinfo

The Net-Net system places a cookie in the `userinfo` portion of a SIP URI when a SIP header contains a SIP URI, and includes that header type in the list of headers to be hidden (encrypted) in the associated SIP NAT function. The cookie for the `userinfo` portion is the following:

```
[user nat tag] [encrypted 13-byte host IP] [encrypted 13 byte maddr IP (if present)]
```

where:

- `[user nat tag]` refers to the SIP NAT function's original user NAT tag field.
- `[encrypted 13-byte host IP]` refers to the host IP encryption.
- `[encrypted 13 byte maddr IP (if present)]` refers to the maddr IP encryption, if it exists.

With a user NAT tag of `-acme`, the following SIP-URI:

```
sip:6175551212@192.168.1.100
```

might be translated into:

```
sip:6175551212-acme-pfi1s7n2pstna@172.16.1.10
```

Note: Multiple additional cookies might be appended with each hop (for example, from the external proxy to the home proxy and back).

host

When hiding IP addresses in a SIP message, the SIP NAT function generates the following cookie for a SIP-URI with no `userinfo` portion:

```
[host nat tag] [encrypted 13-byte host IP] [encrypted 13 byte maddr IP (if present)] [domain suffix]
```

where:

- `[host nat tag]` refers to the SIP NAT function's host NAT tag.
- `[encrypted 13-byte host IP]` refers to the host IP encryption.
- `[encrypted 13 byte maddr IP (if present)]` refers to the maddr IP encryption, if it exists.
- `[domain suffix]` refers to the SIP NAT function's domain suffix field.

With a SIP NAT function's `host tag` of `ACME-` and a domain suffix of `.acme.com`, the following SIP header:

```
Via: SIP/2.0/UDP 192.168.1.100:5060
```

might be translated into the following:

```
Via: SIP/2.0/UDP ACME-pf1ls7n2pstna.acme.com
```

URL Parameter

If the SIP NAT function's `use url parameter` field has a value of `from-to` or `all`, the SIP NAT function places all cookies generated to hide SIP URIs in a custom tag appended to the header. Setting the `use url parameter` field to:

- `from-to` only affects the behavior of the SIP NAT function's cookies in the `From` and `To` headers.
- `all` affects all SIP headers processed by the SIP NAT function

The cookie is the following:

```
[;url-parameter]=[host nat tag] [encrypted 13-byte host IP] [encrypted 13-byte maddr IP]
```

where:

- `[;url-parameter]` refers to the SIP NAT function's parameter name field.
This cookie type is associated with the `all` and `from-to` field value options of the SIP NAT function's `use url parameter` field.
- `[host nat tag]` refers to the SIP NAT function's `host NAT tag` field.
- `[encrypted 13-byte host IP]` refers to the host IP encryption.
- `[encrypted 13 byte maddr IP (if present)]` refers to the `maddr IP` encryption, if it exists.

With a `host NAT tag` of `ACME-` and a parameter name of `acme_param`, the following SIP-URI:

```
sip:6175551212@192.168.1.100
```

might be translated into the following:

```
sip:6175551212@172.16.1.10;acme_param=ACME-pf1ls7n2pstna.
```

tel URL

The SIP NAT function cookie is used when devices in your network are strict about the context portion of SIP messages regarding the conversion of tel URLs. This cookie for the tel URL parameter portion of a SIP message is the following:

```
"tel" URL parameter-[13-byte host IP] [13 byte optional maddr IP] [domain suffix]
```

where:

- `tel URL parameter` refers to the SIP NAT function's `use url parameter`.
This cookie type is associated with the `use url parameter`'s `phone` field value for the SIP NAT.

- [13-byte host IP] refers to the host IP encryption.
- [13 byte optional maddr IP] refers to the maddr IP encryption, if it exists.
- [domain suffix] refers to the SIP NAT function's domain suffix field.

Configuration Overview

Configuring the SIP NAT function falls into two areas, the SIP NAT interface parameters and the SIP NAT policies.

SIP NAT Interface

The following tables lists the SIP NAT function interface parameters you need to configure.

Parameter	Description
realm ID	Name of the external realm. The realm ID must be unique; no two SIP NATs can have the same realm ID. This realm ID must also correspond to a valid realm identifier entered when you configured the realm.
external proxy address	IP address of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net system communicates. Entries must follow the IP address format.
external proxy port	UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net system communicates. Minimum value is 1025, and maximum value is 65535. Default is 5060.
external address	IP address on the media interface in the external realm. Enter a value that ensures any packet with an <code>external address</code> value as its destination address is routed to the Net-Net system through the media interface connected to or routable from the external realm. Entries must follow the IP address format. To specify whether the external realm referenced in this field is private or public, configure the SIP config's NAT mode.
home address	IP address on the media interface in the home realm. Enter a value that ensures any packet with a <code>home address</code> value as its destination address must be routed to the Net-Net system through the media interface connected to or routable from the home realm. Entries must follow the IP address format. The value entered in this field must be different from the IP address value of the home realm's network interface element. The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.

Parameter	Description
home proxy address	Sets the IP address for the home proxy (from the perspective of the external realm). By default, this field is empty. An empty home proxy address field value signifies that there is no home proxy, and the external address will translate to the address of the Net-Net 9200's SIP proxy. Entries must follow the IP address format.
home proxy port	Sets the port number for the home realm proxy. Value can be set to zero (0). Minimum is 1025 and maximum is 65535. Default is 5060.
route home proxy	Whether to route all inbound requests for the SIP NAT to the home proxy. - enabled adds route if Request-URI is not the Net-Net 9200 - disabled does not route inbound requests to the home proxy - forced always adds route

SIP NAT Function Policies

The following tables lists the SIP NAT function policy parameters you need to configure.

Parameter	Description
domain suffix	Domain name suffix of the external realm. The domain name suffix refers to and must conform to the <code>hostname</code> part of a URI. In combination with the <code>user NAT tag</code> and <code>host NAT tag</code> values, this value is used to help the Net-Net system identify an encoded URI that it needs to translate when moving between public and private realms. This suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is <code>ACME-abc123</code> and the domain-suffix value is <code>.netnetsystem.com</code> , the resulting FQDN will be <code>ACME-abc123.netnetsystem.com</code> .
address prefix	Defines which IP address prefixes from incoming messages require SIP-NAT encoding (regardless of the realm from which these messages came).
tunnel redirect	Controls whether Contact headers in a 3xx Response message received by the Net-Net 9200 are NATed when sent to the initiator of the SIP INVITE message.
use url parameter	Establishes whether SIP headers will use the URL parameter entered in the parameter name for encoded addresses that the SIP NAT function creates. Also, if SIP headers will be used, which type of headers will use the URL parameter. For example, all headers or just the From and To headers. Enumeration field.
parameter name	Indicates the name of the URL parameter when use url applies. This field value will be used in SIP NAT encoding addresses that have a use url parameter value of either <code>from-to</code> or <code>all</code> .

Parameter	Description
user NAT tag	Identifies the prefix used when an address is encoded into the username portion of user@host;name=xxxx; where name = parameter name. The user NAT tag values can consist of any characters that are valid for the userinfo part of a URI. In combination with the domain suffix and host NAT tag field values, this value is used to help the Net-Net system identify an encoded URI that it needs to translate when moving between public and private realms.
host NAT tag	Identifies the prefix used when encoding an address into the hostname part of the URI or into a URL parameter. The host NAT tag values refer to domain labels and can consist of any characters that are valid for the hostname part of a URI. In combination with the domain suffix and user NAT tag values, this value is used to help the Net-Net system identify an encoded URI that it needs to translate when moving between public and private realms.
headers	Lists the SIP headers to be affected by the Net-Net system's SIP NAT function. The URIs in these headers will be translated and encrypted, and encryption will occur according to the rules of this SIP NAT.

ACLI Instructions and Examples

To configure the SIP NAT function:

- In Superuser mode, type **configure terminal** and press <Enter>.


```
ACMEPACKET# configure terminal
```
- Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
- Type **sip-nat** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-nat
ACMEPACKET(sip-nat)#
```
- realm-ID**—Enter the name of the realm you want to identify as the external realm.

The name you use as the realm ID must be unique. No two SIP NAT functions can have the same realm ID. Also, this value must correspond to a valid identifier entry already configured for the realm.
- domain-suffix**—Enter the domain suffix to identify the domain name suffix of the external realm. The domain suffix must begin with a (.) dot.

The domain name suffix refers to and must conform to the hostname part of a URI. For example: .netnetsystem.com.

The domain suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is ACME-abc123, the resulting FQDN is ACME-abc123.netnetsystem.com.
- external-proxy-address**—Enter the external proxy address to identify the IP address of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net system communicates.

Enter the value in the IP address format. For example: 192.168.11.200.

7. **external-proxy-port**—Enter the external proxy port value to identify the UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net system communicates. The available values include:

- minimum: 1025
- maximum: 65535

The default value is 5060.

8. **external-address**—Enter the external address, which is an IP address on the media interface in the external realm.

Enter the value in the IP address format. For example:

192.168.11.101

This value must be such that any packet with an `external address` value as its destination address is routed to the Net-Net system through the media interface connected to or routable from the external realm.

9. **home-address**—Enter the home address, which is an IP address on the network interface in the home realm. This value must be such that any packet with a home address value as its destination address must be routed to the Net-Net system through the media interface connected to or routable from the home realm.

Enter the value in the IP address format. For example: 127.0.0.10.

The value entered in this field **must be different** from the IP address value of the home realm's network interface element.

The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.

10. **home-proxy-address**—Enter the home proxy address to set the IP address for the home proxy (from the perspective of the external realm).

By default, this field is empty. No home proxy address entry signifies there is no home proxy, and the external address will translate to the address of the Net-Net system's SIP proxy.

Enter the value in the IP address format. For example: 127.1.0.10.

11. **home-proxy-port**—Enter the home proxy port to set the port number for the home realm proxy.

The available values include:

- zero (0)
- minimum: 1025
- maximum: 65535

The default value is 5060.

12. **route-home-proxy**—*Optional*. Set the route home proxy port to one for the following values:
- enabled: all inbound requests for a specific SIP NAT are routed to the home proxy
 - disabled (default): all inbound requests are not routed through the home proxy.
13. **address-prefix**—*Optional*. Indicate the IP address prefix from incoming messages that requires SIP NAT function encoding (regardless of the realm from which these messages came).

Note: This value overrides the value set in the realm's address prefix field.

This field's format incorporates an IP address and number of bits in the network portion of the address. For example, a Class C address has a 24-bit network part. The address prefix for 101.102.103.x would be represented as 10.102.103.0/24.

If you enter an asterisk (*) or do not enter a value, the realm's address prefix value is used.

14. **tunnel-redirect**—Set to one of the following values to indicate whether certain headers in a 3xx Response message received by the Net-Net 9200 are NATed when sent to the initiator of the SIP INVITE message.
- enabled: certain headers in a 3xx Response message are NATed.
 - disabled (default): certain headers in a 3xx Response message are not NATed.
15. **use-url-parameter**—Establish whether SIP headers will use the URL parameter (configured in the next step) for encoded addresses created by the SIP NAT function. If SIP headers will be used, this value identifies which types of headers will use the URL parameter. The available values include:

- none—No headers will use the URL parameter for address encoding.

The following example illustrates the functionality of a Net-Net system using a use url parameter value of none:

`sip: 1234@1.2.3.4` is translated into `sip:1234-acme-xxxx@5.6.7.8`

where `-acme-xxxx` is a cookie and `xxxx` is the encoded version of `1.2.3.4`.

- from-to—From and To headers will use the URL parameter for address encoding

The following example illustrates the functionality of a Net-Net system using a use url parameter value of none:

`sip:1234@1.2.3.4` is translated into `sip 1234@5.6.7.8;pn=acme-xxxx`

where `-acme-xxxx` is a cookie and `xxxx` is the encoded version of `1.2.3.4`.

- all—All headers will use the URL parameter for address encoding. Acme Packet recommends not using this values because other SIP elements or implementations (other than the Net-Net 9200) might not retain the URL parameter in subsequent SIP messages that they send to the Net-Net 9200.

If this field is set to either `from-to` or `all`, the Net-Net 9200 puts the encoded address of the SIP NAT into a URL parameter instead of using the encoding name inside the userinfo part of the address.

By default, this field is set to `none`, which indicates that Net-Net system functionality remains unchanged.

16. **parameter-name**—If you have configured the user url parameter with the `from-to` or `all` value, you need to indicate the hostname prefix.

The parameter name value is used in SIP NAT encoding addresses that have the use url parameter values of `from-to` or `all`.

17. **user-NAT-tag**—Enter a value to identify the username prefix used for SIP URIs. The values you can use can include any characters valid for the userinfo part of a URI. This should be made unique for each realm and SIP NAT function.

The default value is `-acme-`.

In combination with the domain suffix and host NAT tag values, this value is used to help the Net-Net system identify an encoded URI that it needs to translate when moving between public and private realms.

18. **host-NAT-tag**—Enter a value for the host NAT tag field to identify the hostname prefix used for SIP URIs. The value refers to domain labels and can include any characters valid for the hostname part of the URI. This should be made unique for each realm and SIP NAT function.

The default value is `ACME-`.

In combination with the domain suffix and user NAT tag values, this value is used to help the Net-Net system identify an encoded URI that it needs to translate when moving between public and private realms.

19. **headers**—List the SIP headers you want affected by the SIP NAT function. The URIs in these headers are translated and encrypted, and encryption occurs according to the SIP NAT function rules.

To enter the full default list, type `headers`, followed by a <Space> and `-d`, then press <Enter>.

- 19a. You can also insert the following tags in SIP NAT headers if you want to replace FQDNs with next hop or SIP interface IP addresses:

- `fqdn-ip-tgt`: replaces the FQDN with the target address
- `fqdn-ip-ext`: replaces the FQDN with the SIP NAT external address

Enter the tag using the following format:

`<header-name>=<tag>`

For example:

`To=fqdn-ip-tgt`

The FQDN in a `To` header is replaced with the target IP address.

- 19b. You can insert the following tags to apply NAT treatment to a `From` header in an INVITE when the gateway sends it into the home realm.

- `ip-ip-tgt`: replaces any IP address in the `From` header with the next hop target
- `ip-ip-ext`: replaces any IP address in the `From` header with the Net-Net 9200's external address

- 19c. You can use the **add** and **delete** commands when you enter the **headers** parameter to edit the headers list; entering a new list of values without the **add** or **delete** commands overwrites the entire list.

`ACMEPACKET(sip-nat)# headers add To=fqdn-ip-tgt`

`ACMEPACKET(sip-nat)# headers delete To=fqdn-ip-tgt`

To view all SIP NAT function parameters, enter a ? at the system prompt. The following example shows SIP NAT configuration for peering network.

```

sip-nat
    realm-id                peer-1
    domain-suffix           .p1.acme.com
    ext-proxy-address       192.168.11.200
    ext-proxy-port         5060
    ext-address             192.168.11.101
    home-address            127.0.0.10
    home-proxy-address      127.1.0.10
    home-proxy-port        5060
    route-home-proxy       enabled
    address-prefix          *
    tunnel-redirect         disabled
    use-url-parameter       none
    parameter-name
    user-nat-tag            -p1-
    host-nat-tag            P1-
    headers                 Call-ID Contact From Join
                           Record-Route Refer-To
                           Replaces Reply-To
                           Route To Via

```

SIP Realm Bridging

This section explains how to configure the internal routing among realms known as realm bridging. Realm bridging lets you cross-connect SIP interfaces. You can use one of the following two methods for bridging realms:

- Local policy bridging—Use this method to enable dynamic internal routing between realms if your SIP interfaces do not have the SIP NAT function applied.
- SIP NAT bridging—Use this method if your SIP interfaces have the SIP NAT function applied.

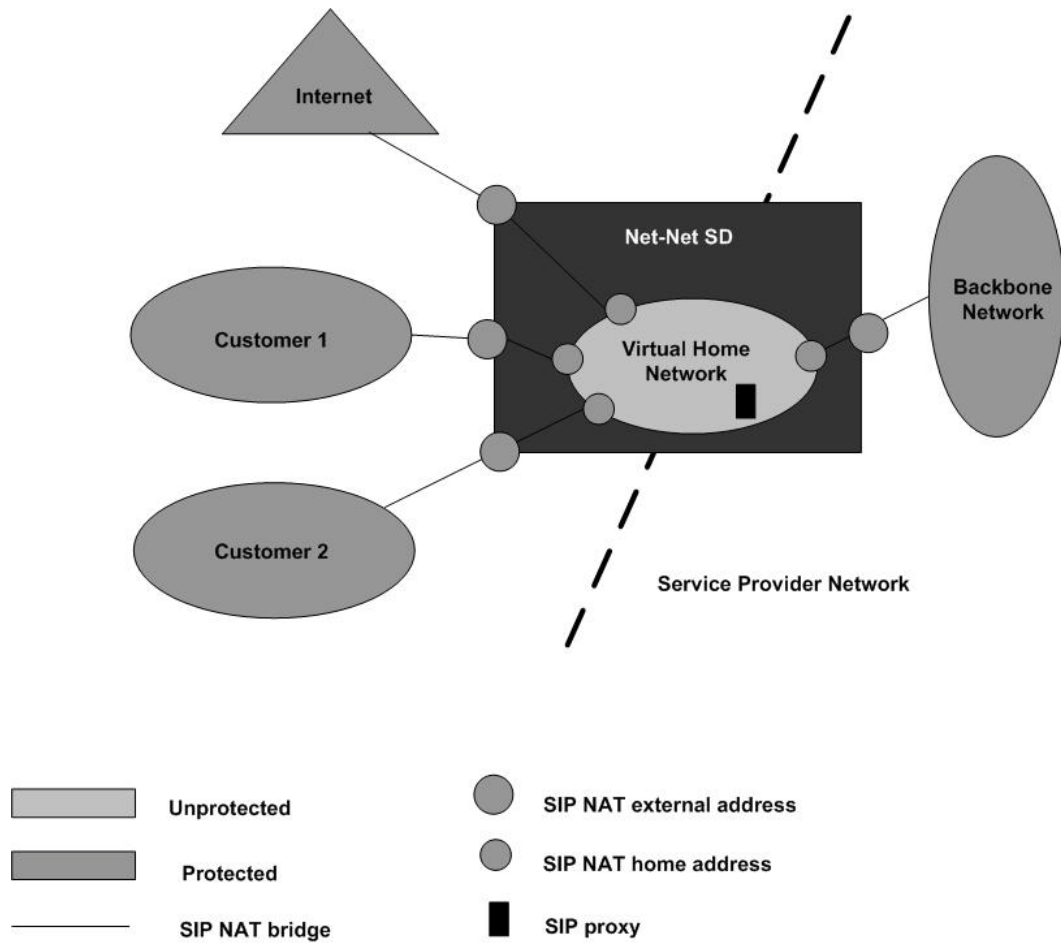
About SIP NAT Bridging

Each SIP NAT has a presence in two realms, trusted and untrusted. The SIP NAT bridge is the conduit for messages in and out of the home realm. It creates a bridge between realms by providing address translations; removing all references to the original IP addressing from the packets sent to the destination network.

With the SIP NAT bridge, an untrusted (or public) home network can reside within the Net-Net 9200, while the other entities (the backbone network, the Internet, or customer networks) are all trusted (or private). One of the primary functions of the SIP NAT bridge is to protect networks from one another so that address bases can remain hidden. Using a SIP NAT bridge, no one network has direct access to the data of other networks.

Establishing a SIP NAT bridge lets you route every SIP Request message through the backbone. Without using this functionality, it would appear as though all messages/sessions were coming from the Net-Net system's SIP proxy (the SIP server that receives SIP requests and forwards them on behalf of the requestor).

The following diagram illustrates this unprotected (or public) and protected (or private) division.



SIP NAT Bridge Configuration Scenarios

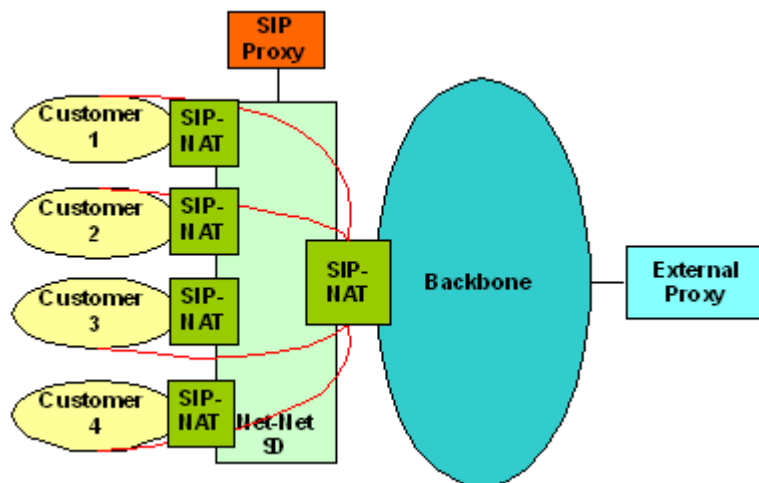
You can configure the SIP NAT bridge functionality in a many-to-one or a one-to-one relationship. For example, multiple customer SIP NATs can be tied to a single backbone SIP NAT, or a single customer SIP NAT can be tied to a single backbone SIP NAT.

You might need to use several SIP NATs on the customer side while using only one on the backbone side in a many-to-one relationship. Or you might configure one SIP NAT on the backbone side for every one that you configure on the customer side in a one-to-one relationship.

You can route all customer side SIP NAT requests to the corresponding backbone SIP NAT regardless of the Request URI.

Many to One Configuration

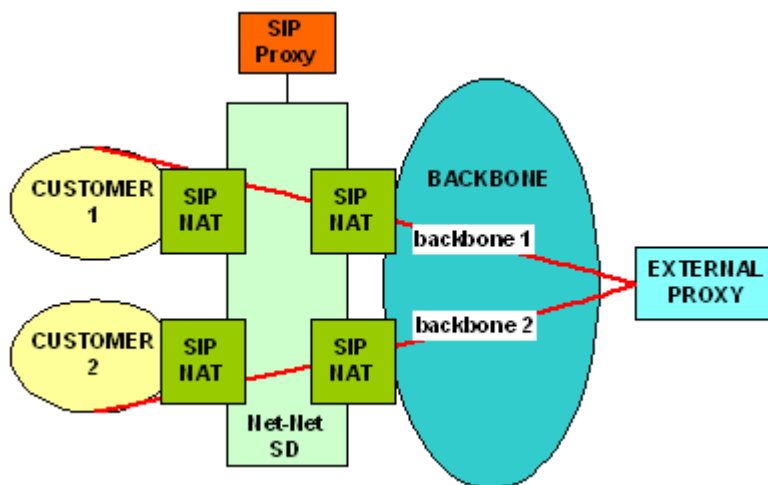
In the many-to-one scenario, multiple customer SIP NATs are tied to a single backbone SIP NAT. The following diagram illustrates the many-to-one SIP NAT bridge configuration.



One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. On the backbone SIP NAT side, you configure the home proxy address to match the home address of the customer SIP NAT. On the customer side, you configure the home proxy address to match the home address of the backbone SIP NAT.

The following diagram illustrates the one-to-one SIP-NAT bridge configuration.



SIP NAT Bridge Configuration

You create a bridge between SIP NATs by pointing them at one another. You point the SIP NATs at each other by configuring the home address and home proxy address to create the bridge. In addition, you can configure the route home proxy on the customer's side of a SIP NAT to force all requests to be routed to the corresponding backbone SIP NAT, regardless of the Request URI. You need to force requests when elements in the customer's network send requests with a Request URI that does not match the customer's SIP NAT external address. Or when the

Request URI does not match a local policy element that would route the requests to the backbone SIP NAT.

You also need a home network to create a SIP NAT bridge. If you do not have a real home network, you need to create a virtual one. You also need to configure instances of the SIP NAT to create the SIP NAT bridge within your network.

Creating a Virtual Home Network

A virtual home network is a home network that resides entirely within the Net-Net 9200, as does a real home network. The difference between the two is the real home network also has a physical connection to the Net-Net 9200.

The internal home realm/network is usually configured with addresses within the special loopback range (127.0.0.0/8) as described in RFC 3330. This applies to the SIP port addresses for the home realm's SIP interface, and all home addresses for SIP NATs. The address 127.0.0.1 should not be used because it conflicts with the default loopback interface setup by the system for inter-process communication.

To create a virtual home network:

1. Set the name and subport ID of the network interface associated with the home realm element to `lo0:0`.
2. To enable the SIP proxy to listen for messages on the virtual home realm, configure the home realm ID. It must correspond to the realm's `identifier`, in which you set the `network interface` subelement to point to the appropriate `network interface` element.

The following table lists the field values you need to set when you are using SIP NAT bridge functionality and you do not have a real home network.

Configuration Element		Sample Values
realm configuration	identifier	acme
	network interfaces	lo0:0
	address prefix	127.0.0.0/8
SIP configuration	home realm ID	acme
	SIP ports address	127.255.255.254

Many-to-One Configuration

To configure many-to-one:

1. For the backbone SIP NAT, ensure the `home proxy address` field is blank.
2. For the customer side SIP NAT:
 - 2a. Set the `home address` to match the home address of the customer.
 - 2b. Set the `home proxy address` to match the backbone SIP NAT home address.

The following table lists the field values you need to set to create a many-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IP address of the home realm. For example: 127.0.0.120
	home proxy address	IP address of the home proxy from the perspective of the external realm. For a backbone SIP NAT, leave blank.
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110

One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. The `home proxy address` field value of the backbone SIP NAT must match the home address of the customer SIP NAT. On the customer side, the home address of the customer SIP NAT should be defined as the home address of the customer, the `home proxy address` field value should match the home address of the backbone SIP NAT.

The following table lists the field values you need to set to create a one-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IP address of the home realm. For example: 127.0.0.110
	home proxy address	IP address of the home proxy from the perspective of the external realm. 127.0.0.120
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110

Shared Session Agent

Usually, the same set of servers (the external proxy) is used for all SIP NATs to the backbone network. In order to support redundant servers in the backbone of a SIP NAT bridge, the original egress realm as determined by the incoming Request URI needs to be retained after a `local policy` lookup.

When a request arrives at the Net-Net 9200, it determines the matching (target) `session agent` and, after the local policy is examined, sets the new outbound `session agent` to the one from the selected target.

If the target session agent's realm is set to `*`, the Net-Net 9200 retains the original egress realm as determined by the SIP NAT or local policy configuration, as applicable.

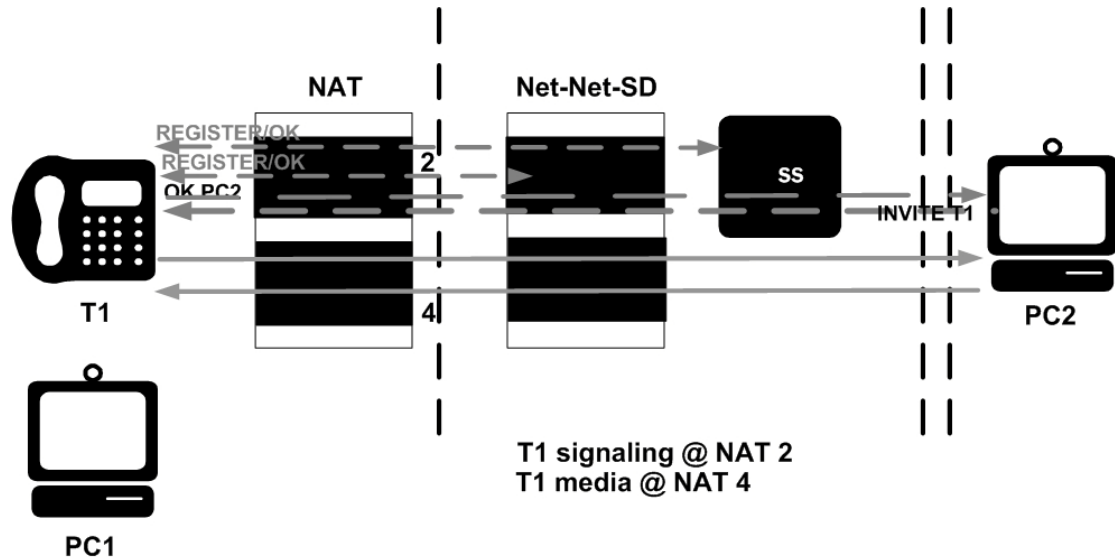
SIP Hosted NAT Traversal (HNT)

This section explains how to configure Hosted Network Address Translation (HNT) traversal. HNT lets endpoints (any system detected behind a NAT) behind a NAT/firewall device send and receive signaling and media using the Net-Net 9200 as a relay.

About HNT

Hosted NAT Traversal (HNT) lets SIP endpoints behind Network Address Translators (NATs) send and receive signaling and media using the Net-Net 9200 as a relay. HNT does not require support for the NAT in the SIP endpoint.

The following diagram illustrates SIP HNT traversal.



The Net-Net 9200's HNT function allows endpoints located behind NATs to communicate; providing means to traverse NATs. The Net-Net 9200 interacts with endpoints (using SIP) to allow persistent inbound and outbound signaling and media communications through these NATs.

The Net-Net 9200 automatically detects when an intermediate NAT exists between the UA and the Net-Net 9200 by comparing the Layer 3 IP address of a REGISTER message with the IP address indicated within the message. The Net-Net 9200 sends signaling responses to the address and port that the request came from, rather than the address and port indicated in the request.

Using HNT with Existing NAT Device

For network architectures in which premise devices and endpoints reside behind an existing NAT device, the Net-Net 9200's HNT function allows these premise NATs to be traversed without requiring an upgrade to the premise equipment, the deployment and management of additional premise-based hardware or software, or any NAT device configuration changes.

Registering Endpoints

The Net-Net 9200 uses periodic endpoint registration messages to establish and maintain bindings in the NAT dynamically. These bindings keep a signaling port (a port that is opened on a firewall to allow traffic to pass through it is a pinhole) open in the NAT, allowing the inbound signaled communications to pass through. Using the endpoint registrations, the Net-Net 9200 then maps the Layer 3 (OSI network layer that deals with switching and routing technologies for data transmission between network devices) IP address/port information from the NAT device to the Layer 5 (OSI session layer that deals with session and connection coordination between applications) entity (for example, user name or phone number) behind the NAT so that when an incoming signaling message is received, the Net-Net 9200 sends it to the appropriate address and port on the NAT for the called party.

Establishing Media Flows

During call setup, the ports for bidirectional media flows are established dynamically. Since the media flows also pass through the Net-Net 9200, it can identify the IP address/port information on the NAT device used for the outgoing media coming from the user name/phone number. The Net-Net 9200 then uses that same NAT's IP address/port information to send incoming media to the correct user name/phone number behind the NAT device.

Prerequisites

In order to achieve HNT, the endpoints involved must be capable of:

- Symmetric signaling—Sending and receiving SIP messages from the same transport address (IP address or User Datagram Protocol/Transmission Control Protocol (UDP/TCP) port)
- Symmetric media—Sending and receiving Real-Time Transport Protocol (RTP) messages from the same UDP port

These conditions are required to allow signaling and media packets back through the NAT (through the bound external address and port). These packets must come from the address and port to which the outbound packet that created the NAT binding was sent. The NAT sends these inbound packets to the source address and port of the original outbound packet.

When SIP HNT is used, the Net-Net 9200 sends signaling responses to the address and port that the request came from rather than the address and port indicated in the request. These are added to the topmost Via within the `received=` and `rport` parameters, respectively.

Keeping the NAT Binding Open

Additional measures are also required to keep the NAT binding open because most NAT bindings are discarded after a few minutes of inactivity. The Net-Net 9200 keeps the SIP NAT binding open by returning a short expiration time in REGISTER responses that forces the endpoint to send frequent REGISTER requests.

In order to keep the NAT binding open for SIP, the Net-Net 9200 maintains the registration state. When an endpoint first registers, the Net-Net 9200 forwards that REGISTER message on to the real registrar. You can define the real registrar using either of the following methods:

- Configure the SIP config registrar host and registrar port to indicate the real registrar.

- Map the SIP config registrar host and registrar port values to the SIP NAT home proxy address and home proxy port values. Then configure the SIP NAT's external proxy address and external proxy port values to correspond to the real registrar.

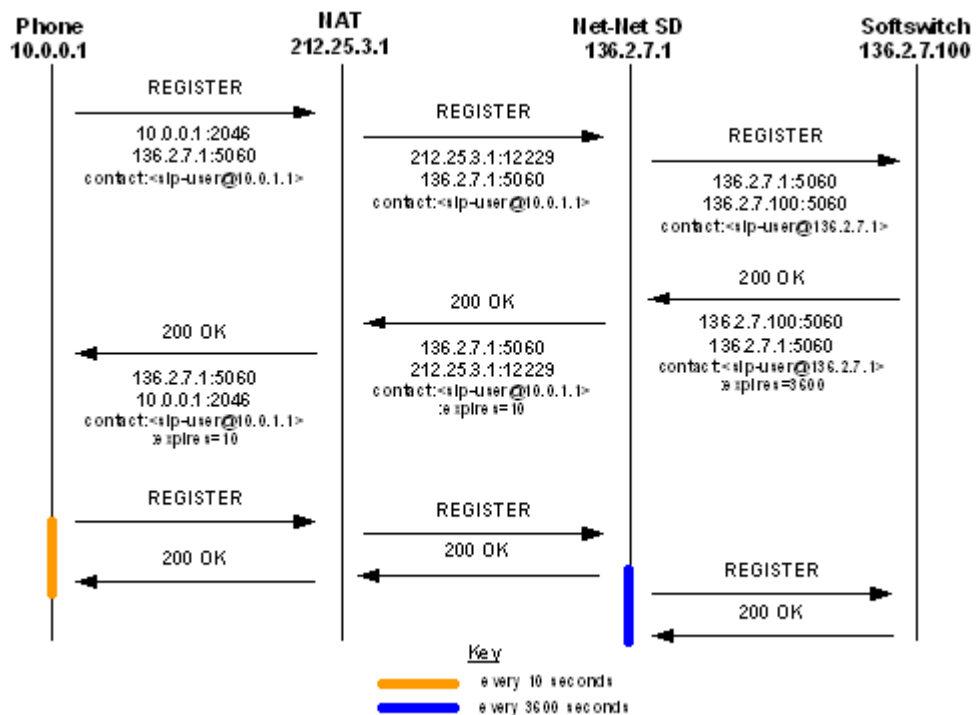
Note: A registrar can be located in a SIP NAT realm.

When a successful response is received, the Net-Net 9200 caches the registration to memory. This cached registration lives for the length of time indicated by the expiration period defined in the REGISTER response message from the registrar. The response sent back to the endpoint has a shorter expiration time (defined by the SIP interface's NAT interval) that causes the endpoint to send another REGISTER message within that interval. If the endpoint sends another REGISTER message before the cached registration expires, the Net-Net 9200 responds directly to the endpoint. It does not forward the message to the real registrar.

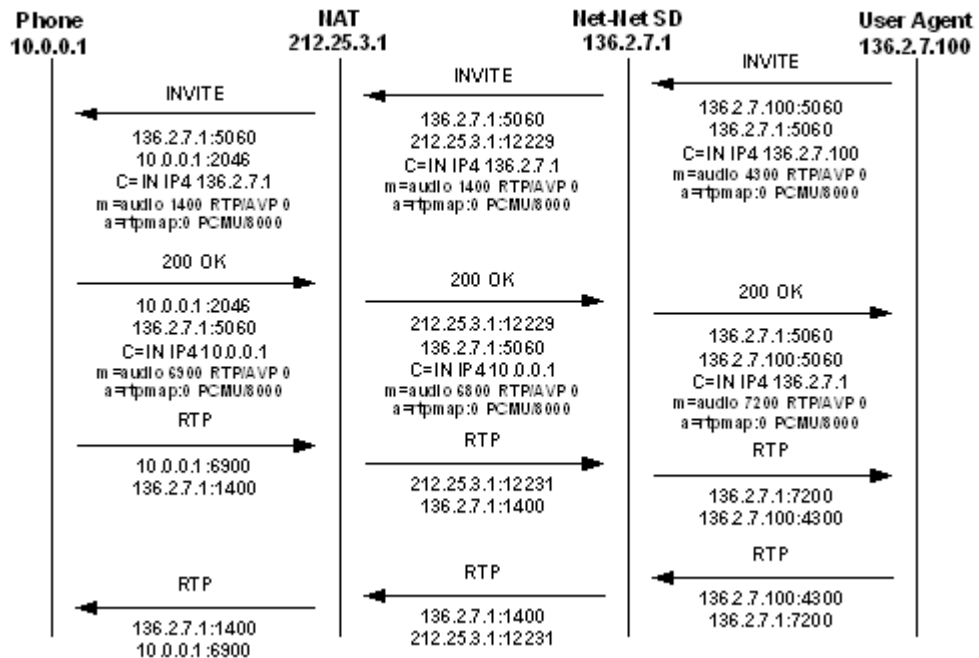
If the cached registration expires within the length of time indicated by the NAT interval, the REGISTER message is forwarded to the real registrar. If the Net-Net 9200 does not receive another REGISTER message from the endpoint within the length of time indicated by the NAT interval, it discards the cached registration.

The Contact Uniform Resource Identifier (URI) in the REGISTER message sent to the registrar by the Net-Net 9200 points at the Net-Net 9200 so that the proxy associated with the real registrar sends inbound requests to the Net-Net 9200. This way, the inbound requests can be forwarded to the endpoint through the NAT binding.

The following example illustrates the SIP HNT registration call flow for the SIP HNT feature.



The following example illustrates the SIP HNT invitation call flow for the SIP HNT feature.



Working with Multiple Domains

You can use a wildcard (*) with the HNT feature to accommodate multiple domains and to allow the Net-Net 9200 to cache all HNT endpoints. The wildcard functionality is enabled in the `SIP config` by entering an asterisk (*) in the registrar domain and registrar host fields.

The wildcard allows the use of either a local policy or Domain Name Service (DNS) to resolve the domain name to the correct registrar. Either method can be used to route the Fully Qualified Domain Name (FQDN) when the you enter an asterisk (*) for the register host. An FQDN consists of an unlimited number of domain labels (domain names), each separated by a dot (.). The FQDN can include the top level domain name (for example, `acmepacket.com`).

In the hostname `acme-packet.domainlbl.example100.com`, the syntax is as follows:

- `acme-packet` is a domain label
- `domainlbl` is a domain label
- `example100` is a domain label
- `com` is the top label

The information configured in a local policy is used before DNS is used. If the next hop destination address (defined in the local policy's `next_hop` parameter) is an IP address, a DNS server is not needed. A DNS server is needed when the IP address of the next hop destination address is a FQDN or cannot be determined from the Net-Net 9200's configuration. Even with a configured `local_policy`, the next hop destination address might be an FQDN that requires a DNS lookup.

If the registrar host does not use the wildcard, the Net-Net 9200 always uses the configured address. You can limit the number of endpoints that receive the HNT function. For example, you can use a non-wildcarded `registrar domain` field value (like `acme.com`) with a wildcarded registrar host field value.

TCP NAT Interval

The Net-Net SBC supports hosted NAT traversal for UDP and TCP. Because endpoints that register over TCP do not need to send refresh requests as frequently as those registering over UDP, a lower system load can be attained if registration over is known. By adding a separate configuration for the TCP NAT interval, the load is reduced.

How It Works

The Net-Net SBC uses the value you set for the TCP NAT interval as the expiration value passed back in SIP REGISTER (200 OK) responses to endpoints behind a NAT that register over TCP. The NAT interval value with which you are familiar from previous releases is used for endpoints behind a NAT that register over UDP.

HNT Configuration Overview

To configure SIP HNT NAT traversal, you need to configure both the SIP interface and the SIP config.

SIP HNT Single Domain Example

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a single domain and registrar.

- SIP config

Parameter	Sample Value
registrar domain	netnetsystem.com
registrar host	192.168.12.1
registrar port	5060

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60
TCP NAT interval (optional)	90
minimum registration expire	300
registration caching	disabled
route to registrar	enabled

SIP HNT Multiple Domain Example

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a multiple domains and multiple registrars.

- SIP config

Parameter	Sample Value
registrar domain	*
registrar host	*

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60
minimum registration expire	300
registration caching	disabled
route to registrar	enabled

ACLI Instructions and Examples

This section provides instructions for how to configure your Net-Net 9200 for HNT.

SIP Interfaces and HNT

To configure a SIP interface for HNT:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

From this point, you can configure physical interface parameters. To view all SIP interface parameters, enter a ? at the system prompt.
4. **nat-traversal**—Define the type of HNT enabled for SIP. Available values include:
 - **none**—Disables the HNT feature for SIP (default value)
 - **rport**—SIP HNT function only applies to endpoints that include the rport parameter in the Via header. HNT applies when the source IP address of the SIP request does not match the IP address in the Via header.
 - **always**—SIP HNT applies to requests with a source IP address that does not match the IP address in the Via header (even when the rport parameter is not present.)
5. **nat-interval**—set the expiration time in seconds for the Net-Net 9200's cached registration entry for an HNT endpoint. The default is 30, and other available values include:
 - minimum: 0 seconds
 - maximum: 999999999 seconds

Acme Packet recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to prevent the NAT device from keeping an unused port open.

6. **tcp-nat-interval**—Enter the registration cache expiration time in seconds to use for endpoints behind a NAT device that register using TCP. For new SIP interface configurations, this parameter defaults to 90. On upgrade, the Net-Net SBC assigns this parameter the same value as the existing NAT interval.
7. **registration-caching**—Enable for use with all UAs, not just those that are behind NATs. By default, this field is set to disabled. If enabled, the Net-Net 9200 caches the Contact header in the UA's REGISTER request when it is addressed to one of the following:
 - Net-Net 9200
 - registrar domain value
 - registrar host value

The Net-Net 9200 then generates a Contact header with the Net-Net 9200's address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.

Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:

- HNT enabled—The Net-Net 9200 takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.
- HNT disabled—The user part of the Contact header is taken from the URI in the From header and no cookie is appended. This is the default behavior of the Net-Net 9200.

When the registrar receives a request that matches the address-of-record (the To header in the REGISTER message), it sends the matching request to the Net-Net 9200, which is the Contact address. Then, the Net-Net 9200 forwards the request to the Contact-URI it cached from the original REGISTER message.

8. **min-reg-expire**—Set the time in seconds for the SIP interface. The value you enter here sets the minimum registration expiration time in seconds for HNT registration caching.
 - minimum: zero (0) seconds
 - maximum: 999999999 seconds

The default value is 300.

This value defines the minimum expiration value the Net-Net 9200 places in each REGISTER message it sends to the real registrar. In HNT, the Net-Net 9200 caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the Net-Net 9200 to send frequent registrations on to the real registrar.

9. **registration-interval**—Set the Net-Net 9200's cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the Net-Net 9200 to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires.

- minimum: zero (0) seconds

A registration interval of zero causes the Net-Net 9200 to pass back the expiration time set by and returned in the registration response from the registrar.

- maximum: 999999999 seconds

The default value is 3600 seconds (one hour). If the expiration time you set is less than the expiration time set by and returned from the real registrar, the Net-Net 9200 responds to the refresh request directly rather than forwarding it to the registrar.

Note: With registration caching, there is no NAT; therefore, a short registration interval causes the UA to send excess REGISTER messages.

Although the registration interval applies to non-HNT registration cache entries, and the loosely-related NAT interval applies to HNT registration cache entries, you can use the two in combination. Using a combination of the two means you can implement HNT and non-HNT architectures on the same Net-Net 9200. You can then define a longer interval time in the registration interval field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

10. **route-to-registrar**—Enable routing to the registrar to send all requests that match a cached registration to the destination defined for the registrar host; used when the Request-URI matches the registrar host value or the registrar domain value, not the Net-Net 9200's address. Because the registrar host is the real registrar, it should send the requests back to the Net-Net 9200 with the Net-Net 9200's address in the Request-URI. By default, this field is set to `disabled`.

For example, you should enable routing to the registrar if your network uses a Net-Net 9200 and needs requests to go through its service proxy, which is defined in the registrar host field.

The Global SIP Configuration and HNT

To configure the global SIP configuration for HNT:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a `?` at the system prompt.
4. **registrar-domain**—*Optional*. Define the domain to match against the host part of a URI to determine if a request is addressed to the registrar. If there is a match, the registration caching, NAT traversal, and route to registrar parameter values for the SIP interface are applied to the request. By default, this field remains empty. Available values are:
 - An asterisk (*) to specify the values apply to all requests.
 - Any alphanumeric character or any combination of alphanumeric characters. For example, `acme1.com`.

A hostname consists of any number of domain labels, separated by dots (.), and one top label. A top label is the last segment of the hostname. It must start with an alphabetical character. After the first character, a top label can consist of any number or combination of alphanumeric characters, including those separated by dashes. The dash must be preceded and followed by alphanumeric characters. A single alphabetical character is the minimum requirement for a hostname field (for example, `c` to indicate `.com`).

When the REGISTER message's Request-URI has an FQDN, it is matched against the registrar domain's value to determine if the message needs to be forwarded to the registrar port on the registrar host. The registrar domain's value is also used when route to registrar is set to `enabled`, to determine if a request needs to be forwarded to the registrar.

Only the right-hand part of the domain name in the Request-URI needs to match the registrar domain value. For example, `acme3.acmepacket.com` matches `acmepacket.com`. However, the entire domain label within the domain name must match. For example, the domain label "acme3.acmepacket.com" would not match "packet.com".

5. **registrar-host**—Define the address of the registrar for which requests for registration caching, NAT traversal, and router to registrar options apply. You can use a specific hostname, a IP address, or a wildcard (*):
 - An asterisk (*)—Indicates normal routing (local policy, DNS resolution, and so on) is used to determine the registrar's address.
 - Hostname—Can consist of any alphanumeric character or any combination of alphanumeric characters (for example, `acme1.com`). The hostname can consist of any number of domain labels, separated by dots (.), and one top label. You can use the minimum field value of a single alphabetical character to indicate the top label value (for example, `c` to indicate `.com`).
 - IP address—Must follow the dotted notation format. Each of the four segments can contain a numerical value between zero (0) and 255. For example, `192.168.201.2`. An example of a invalid segment value is `256`. See the *ACLI User Interface* chapter of the *Net-Net ACLI Reference Guide* for more information about entries in IP address fields.

By default, the registrar host field remains empty.

6. **registrar-port**—Set the SIP registrar port number. The SIP registrar server configured in this and the registrar host field is the real registrar. Or the values entered in those fields map to the home proxy address and home proxy port of the SIP NAT with external proxy address and external proxy port values that correspond to the real registrar.

Available values include:

- minimum: zero (0) is the default
- range: between 1025 and 65535

The following example shows the values mentioned above for a single domain and registrar configuration.

```

sip-config
    registrar-domain      nn9200.com
    registrar-host        192.168.12.1
    registrar-port        5060
  
```

HNT Forced Unregistration

This section explains how to configure HNT force unregistration. When Net-Net 9200's forced unregistration is enabled and a registration entry for an endpoint

expires, the Net-Net 9200 will notify the softswitch to remove this binding using REGISTER message. In that REGISTER message, the expires header will be set to 0 and the expires parameter in the Contact header will also be set to 0.

The benefits of using forced unregistration include:

- Leveraging existing HNT configuration to provide near real-time information about the UA's status to the registrar/softswitch
- Preserving resource utilization for the Net-Net 9200 and the softswitch by deleting a contact binding that is no longer valid or needed
- Preventing extra bindings from being generated at the softswitch (e.g., in instances when the UA or NAT restart)

When to Use Forced Unregistration

For typical HNT use, it is common that registration interval between the client UA and the Net-Net 9200 is between sixty (60) and one hundred and twenty (120) seconds. This differs significantly from the re-registration interval between the Net-Net 9200 and the registrar, which varies from approximately thirty (30) to sixty (60) minutes.

If the UA fails to refresh its registration (for any possible reason), the contact binding at the Net-Net is deleted after the registration expires. This expiration is determined by the expires= header in the 200 OK. However, the binding at the real registrar will remain intact. This creates a discrepancy between the real state of the UA and state at the softswitch. In the best case scenario, the contact binding expires at the softswitch after a few minutes.

From the perspective of network management, this discrepancy can be problematic because the service provider would be unaware of the UA's status until the binding expires at the softswitch. This can take a considerable amount of time to happen.

In addition, the Net-Net 9200 encodes a cookie in the userinfo of the Contact header in the REGISTER message. This is a function of the source IP address and port from which the request came, i.e., the ephemeral port in the NAT for DSL scenarios. Therefore, additional bindings that remain for long periods of time are created at the registrar if, for example, the:

- UA reboots
- Ethernet link between the UA and the DSL router is lost for over two minutes
- DSL crashes
- DSL/ATM layer between the DSL router

Caution for Using Forced Unregistration

You should use caution when applying SIP HNT forced unregistration for the following reasons:

- It can have an impact on the performance of your Net-Net 9200 and the registrar, especially when you have a large number of HNT endpoints in your configuration that become unavailable simultaneously.
- It is possible that the registrar might become vulnerable to overload in the case where the registrar must authenticate a large number of register messages generated when HNT endpoints are de-registered. It is possible that the cached registration credentials might become "stale" over time (e.g., the "nonce" value

usually has a limited lifetime). Without proper credentials, the registrar will reject the de-registrations.

Given these concerns, we recommend that you consult with your Acme Packet systems engineer before adopting the use of forced unregistration.

Configuring Forced Unregistration

This feature applies to:

- HNT endpoints with registration caching enabled by default, and when the **nat-traversal** parameter in the SIP interface configuration is set to **always**
- non-HNT endpoints with registration caching enabled, when the registration-interval parameter in the SIP interface configuration is used in the expires header sent to the UA in the 200 OK

ACLI Instructions and Examples

To enable SIP HNT forced unregistration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
```
4. Use the ACLI **select** command so that you can work with the SIP configuration.


```
ACMEPACKET(sip-config)# select
```
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **force-unregistration**, and then press <Enter>.


```
ACMEPACKET(sip-config)# options +force-unregistration
```

If you type **options force-unregistration**, you will overwrite any previously configured options. In order to append the new option to the **sip-config**'s options list, you must prepend the new option with a "plus" sign as shown in the previous example.

Adaptive HNT

This section explains how to configure adaptive HNT. The adaptive HNT expires feature allows the Net-Net SBC to automatically determine the maximum SIP REGISTER message expires time interval in order to keep each individual NAT pinhole open when performing SIP HNT.

Overview

Without adaptive HNT, the Net-Net SBC keeps NAT pinholes open and port mapping cached by forcing the UAC to send frequent SIP REGISTER messages. It does so by setting the expires time to a short interval. Some NATs only need a message to be sent by the private client once every twenty minutes, while other NATs delete their cache/pinhole in thirty seconds if no messages appear. Given this large variation in time intervals, the Net-Net SBC's nat-interval (expire time) has been set to a low value in order to support as many NAT types as possible. However, CPU performance and scalability issues result from such a small refresh time, especially when there is a very large number of potential registered users.

When you use adaptive HNT, the Net-Net SBC waits for a time interval and then sends a SIP OPTIONS message to the UAC to see if it can still be reached. If the UAC can still be reached, the Net-Net SBC increases the timer and tries again. In case the pinhole closes because it has exceeded the NAT's cache time, the Net-Net SBC sets the expires time to be slightly longer than the time it tests using the OPTIONS method. This way, the UAC will send another REGISTER message shortly thereafter and impact on service will be minimal.

Adaptive HNT Example

An example call flow using adaptive HNT involves a basic HNT user and a Net-Net SBC. It begins when the Net-Net SBC receives and forwards the 200 OK for the REGISTER message. Then the Net-Net sends an expires timer for slightly longer than the time for which to test; in this example, it begins the test for the amount of time set for the minimum NAT interval. It adds ten seconds to this time when it sends the expires timer. This way, there is time for the OPTIONS message to be sent before the REGISTER message is received (which would refresh the NAT's cache). The Net-Net SBC also tries to keep the REGISTER time short enough so that even if the NAT pinhole closes, there is minimal time before the UAC creates a new NAT binding by sending another REGISTER. Because a ten second interval may be too long, you might want to set this value to a better-suited time.

The test succeeds with a minimum test-timer because the UAC responded to the OPTIONS message. So the test-timer value is increased by thirty seconds and tried again. The expires time in the REGISTER message will be increased to the test-timer value plus ten seconds. This time, the UAC does not respond to the OPTIONS message even though it was sent multiple times. Because the OPTIONS fails, when the Net-Net SBC receives another REGISTER, it responds with the previously successful timer value (in this case, the minimum NAT interval).

However, if the OPTIONS request succeeds, then the Net-Net SBC persists with the test until it fails or until the maximum NAT timer value is reached. In this case, when the OPTIONS message fails, the Net-Net SBC uses the last successful test-timer value as the time for the expires header in the 200 OK for the REGISTER message.

Synchronize A-HNT Successful Timer to Standby

Adaptive HNT enables the Net-Net SBC to determine, through testing, an optimum SIP REGISTER expires time interval that keeps the NAT pinhole open. For an HA node, this successful time value is determined through testing by the active system and then replicated to the standby. If there is a switchover during the active system's testing process, then it will restart for that endpoint.

ACLI Instructions and Examples

You configure the SIP interface to set the state of this feature and to define the increments of time the Net-Net SBC uses to perform adaptive HNT. Remember that the Net-Net SBC uses the time you specify as the NAT interval, the supported time interval, as the basis on which to begin testing.

To configure adaptive HNT:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
`ACMEPACKET(session-router)# sip-interface`
4. **sip-dynamic-hnt**—Enable this parameter if you want to use adaptive HNT. The default value is **disabled**. The valid values are:
 - enabled | disabled
5. **max-nat-interval**—Set the amount of time in seconds that testing should not exceed. The Net-Net SBC will keep the expires interval at this value. The default value is **3600**. The valid range is:
 - Minimum—0
 - Maximum—999999999
6. **nat-int-increment**—Set the amount of time in seconds to use as the increment in value in the SIP expires header. The default value is **10**. The valid range is:
 - Minimum—0
 - Maximum—999999999
7. **nat-test-increment**—Set the amount of time in seconds that will be added to the test timer. The default value is **30**. The valid range is:
 - Minimum—0
 - Maximum—999999999

SIP Server Redundancy

This section explains how to configure SIP server redundancy. SIP server redundancy involves detecting that an upstream/downstream SIP signaling entity has failed, and adapting route policies dynamically to remove it as a potential destination.

Overview

You establish SIP server redundancy by creating session agents, which are virtual representations of the SIP signaling entities. These agents are then collected into a session agent group, which is a logical collection of two or more session agents that behaves as a single aggregate entity. For more information about session agents and session agent groups, see the Session Routing and Load Balancing chapter in this guide.

Rather than direct signaling messages to a single session agent (IP), the signaling message is directed to a session agent group (SAG). The group will have a set distribution pattern: hunt, round robin, proportionally distributed, and so on. Signaling is spread amongst the agents using this chosen pattern.

You direct the signaling message by configuring a route policy, known as a local policy, which determines where SIP requests should be routed and/or forwarded. The values in the To and From headers in the SIP request are matched with the content of the local policy within the constraints set by the session agent's previous hop value and SIP interface values such as the list of carriers.

To summarize, you need:

- two or more session agents
- a session group containing those session agents
- a local policy which directs traffic to the session agent group

Configuration Overview

You make a session agent group a target by using a local policy to select the next hop from the members of a session agent group. You need to set the `replace URI` field of the configured local policy to `enabled`; which causes NAT rules such as realm prefixing to be overridden. The `replace URI` field allows you to indicate whether the local policy's value is used to replace the Request-URI in outgoing requests. This boolean field can be set to either `enabled` or `disabled`.

When the SIP NAT's `route home proxy` field is set to `forced`, it forces the request to be forwarded to the `home proxy` without using a local policy. When this option is set to either `disabled` or `enabled` and the Request-URI matches the external address of the SIP NAT, the local policy is used.

However, the local policy only replaces the Request-URI when the original Request-URI matches the Net-Net system's IP address or hostname. This behavior is in accordance with that described in RFC 3261. The original Request-URI will be the `home proxy address` value (the home address of the SIP NAT into the backbone) and not the Net-Net system's address.

Using strict routing, the Request-URI would be the next hop, but the message would also include a Route header with the original Request-URI. With loose routing, the Request-URI remains unchanged and the `next hop` value is added as the top Route header.

In some cases, the `next hop` field value must replace the Request-URI in the outgoing request, even if the original Request-URI is not the Net-Net system. To accomplish this, an option has been added to the local policy that causes the `next hop` value to be used as the Request-URI and prevents the addition of Route headers. This option is part of the **action** parameter in the local policy attributes.

The following table lists the policy attributes for the local policy:

Parameter	Description
<code>next hop</code>	IP address of your internal SIP proxy. This value corresponds to the IP address of the network interface associated with the SIP proxy.
<code>realm</code>	Number of the port associated with the SIP port.
<code>action</code>	Stores the transport protocol used for sending an receiving signaling messages associated with the SIP port.

Note: You should also define the ping method intervals for the session agents so that the Net-Net 9200 can detect when the agents are back in service after failure.

For more information about local policy, see the [Session Routing and Load Balancing \(583\)](#) chapter in this guide.

ACLI Instructions and Examples

To enable replace URI:

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


```
ACMEPACKET# configure terminal
```
2. Type **local-policy** and press <Enter> to access the system-level configuration elements. The system prompt changes.


```
ACMEPACKET(configure)# local-policy
ACMEPACKET(local-policy)#
```
3. Type **policy-attributes** and press <Enter>. The system prompt changes.


```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

From this point, you can configure policy attributes for the local policy. To see all local policy attribute options, enter a ? at the system prompt.
4. **replace-uri**—Set the replace URI field of the configured local policy to enabled; which causes NAT rules such as realm prefixing to be overridden.

The replace URI field allows you to indicate whether the local policy's value is used to replace the Request-URI in outgoing requests. This boolean field can be set to either enabled or disabled.

SIP Distributed Media Release (DMR)

This section explains how to configure distributed media release (DMR). SIP DMR lets you choose whether to include multi-system (multiple Net-Net SDs) media release information in SIP signaling requests sent into a specific realm.

Overview

The SIP DMR feature lets RTP/RTCP media be sent directly between SIP endpoints (for example, SIP phones or user agents) without going through a Net-Net 9200; even if the SIP signaling messages traverse multiple Net-Net SDs. It encodes IP address and port information for the media streams described by the media, for example SDP.

With SIP DMR, the media realm and IPv4 address and port information from the UA's SDP is encoded into SIP messages (either in the SIP header or in the SDP) as they enter the backbone network. The information is decoded by a Net-Net 9200 from SIP messages that come from the backbone network. The decoded address and port information is put into the SDP sent the UAs in the access (private/customer) network.

This functionality lets the RTP/RTCP flow directly between the UAs in the access network without traversing the Net-Net SDs and without passing into the backbone network. The media can then flow directly between the two SIP endpoints in the same network, if it is serviced by multiple Net-Net SDs.

You can enable this feature on a per-realm basis and multiple realms can be supported.

Endpoint Locations

You can configure the Net-Net 9200 to release media when the source and destination of the call are in the same network, customer VPN, or customer LAN. In architectures that use DMR, the Net-Net 9200 is only part of the media path for traffic that originates and terminates in different networks.

If configured to do so, the Net-Net 9200 can release media:

- Between endpoints supported by a single Net-Net 9200
 - In the same network/VPN
 - In the same network behind the same NAT/firewall
- Between endpoints supported by multiple distributed Net-Net SDs
 - In the same network/VPN

Location of the Encoded Information

Encoded media release information can appear in three different places:

- SDP attribute

Media release data can be encoded into an SDP attribute in the SIP message body (for example, `media-release=sdp;acme-media`). The encoded data is placed into an `acme-media` attribute in the SDP:

```
a=acme-media:<encoded-media-interface-info>
```

- SIP header parameter

Media release data can be placed in a header parameter of a SIP header (for example, `media-release=Contact;acme-media`). The encoded data is placed into an `acme-media` parameter in the Contact header:

```
Contact: <sip:1234@abc.com>;acme-media=<encoded-media-interface-info>
```

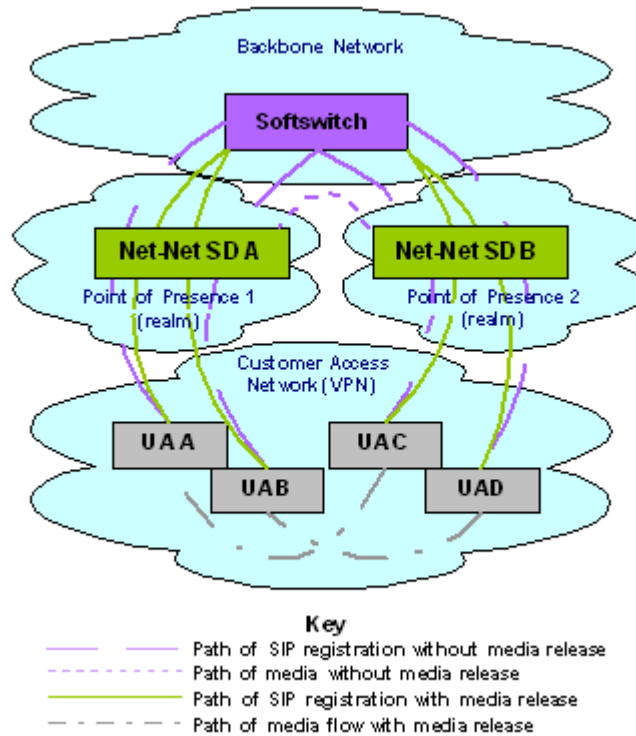
- SIP header

Media release data can appear in a SIP header (for example, `media-release=P-Media-Release`). The encoded data is placed into a `P-Media-Release` header:

```
P-Media-Release: <encoded-media-interface-info>
```

Example: Distributed Media Release

The following example shows the network diagram for DMR in a multiple-site VPN environment supported by multiple, distributed Net-Net SDs.



As shown in the network diagram, UA A and UA B register with the softswitch through Net-Net 9200 A while UA C and UA D register with the softswitch through Net-Net 9200 B. Without DMR, the media for calls between UA A/UA B and UA C/UA D is steered through both Net-Net SD A and Net-Net SD B.

With SIP DMR, the media realm and IPv4 address and port information from the UA's Session Description Protocol (SDP) are encoded into SIP messages (either in the SIP header or in the SDP) as they enter the backbone (public/service provider) network. The information is decoded from SIP messages that come from the backbone network. The decoded address and port information is put into the SDP sent to the UAs in the access (private/customer) network. This functionality allows for the RTP/RTCP to flow directly between the UAs in the access network without traversing the Net-Net SDs and without passing into the backbone network.

Overview of SIP DMR Configuration**To configure SIP DMR:**

1. Edit the SIP config element's option field.

The media-release="`<header-name>[:<header-param>]`" option defines how the SIP distributed media release feature encodes IPv4 address and port information. If the media-release parameter is configured in the options field but no header is specified, the parameter value of P=Media-Release will be used. This parameter is optional and is not configured by default.

2. Enable SIP DMR for the entire realm by setting the realm config element's msm release field to enabled.

The media IPv4 address and port information is encoded into outgoing SIP messages and decoded from incoming SIP messages for all of the realms (in each realm-config element) with which the SIP distributed media release will be used. The realm names/identifiers must be identical.

Note: You can also use the realm config element's `mm in network` field to release the media back to a connected network that has multiple realms. This field is not specific SIP distributed media release and it is not required for the SIP DMR to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Net-Net 9200 release the media.

ACLI Instructions and Examples

To configure media release:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
```

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

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a `?` at the system prompt.

4. Type **options** followed by a <Space>.
5. After the <Space>, type the media release information in the following format:

```
media-release="<header-name>[;<header-param>]"
```

- `header-name` either refers to the SIP header in which to put the information or to the special `header-name` value of `sdp` to indicate the information should be put into the SDP.
- `parameter-name` refers to the header parameter name in which to put the information or, in the case of the **special** `header-name` value of `sdp`, to the SDP attribute name in which to put the information.

For example:

```
ACMEPACKET(sip-config)# options media-release=P-Media-Release
```

6. If you are inserting the DMR cookie into the SDP message, you can specify the SDP attribute name where the MSMR check address cookie is inserted using the following syntax:

```
ACMEPACKET(sip-config)# options +msm-release-check-address=MSMR-cookie-goes-here
```

7. Save your work using the ACLI **done** command.

Note: If the `media-release` parameter is configured in the `options` field, but no header is specified, then the parameter value of `P-Media-Release` will be used. `P-Media-Release` is a proprietary header and means that the media will be encoded in the SIP header with this name.

The following example shows where the encoded information (for example, SDP data) is passed.

```
media-release="P-Media-Release"
media-release="Contact;acme-media"
media-release="sdp;acme-media"
```

Configuring the Realm Configuration

You need to set the realm configuration's **msm-release** parameter to enabled for all the realms for which you want to use SIP DMR.

Although the **mm-in-network** parameter is not specific to the SIP distributed media release feature, it can be used to release the media back to a connected network that has multiple realms. This field does not need to be configured in order for the SIP distributed media release feature to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Net-Net 9200 release the media.

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.
4. **msm-release**—Set this parameter to enabled (default is disabled) to enable DMR within this realm on this Net-Net 9200.
5. Repeat for each realm on which you want to enable DMR.

Embedded Header (E911) Support

This section explains how to configure embedded header support. The Net-Net 9200 supports methods of extracting an embedded P-Asserted-Identity header from a contact header to support E911 when integrated with certain vendor's systems. See RFC 3455 *Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)* for more information.

The embedded header support feature watches for a specified embedded header contained in a Contact header received in a 3XX message. When the specified embedded header is found, the full <header=value> pair is inserted as a unique header in a redirected INVITE message that exits the Net-Net 9200. If the outgoing INVITE message were to contain the specified header, regardless of the use of this feature, the value extracted from the 3XX message replaces the INVITE message's specified header value.

If an incoming Contact header in a 3XX message looks like:

```
Contact: <ESRN@IPv4_Intrado_GW;user=phone?P-Asserted-Identity=%3Csip:+1-ESQK@IPv4_My_EAG;user=phone%3E>
```

when you configure your Net-Net 9200 to parse for the embedded P-Asserted-Identity header to write as a unique header in the outgoing invite message, the outgoing INVITE and P-Asserted-Identity headers will look like:

```
INVITE SIP: ESRN@IPv4_Intrado_GW;user=phone
```

P-Asserted-Identity: +1-ESQK@IPv4_My_EAG;user=phone

ACLI Instructions and Examples

Embedded header support is enabled in the session agent configuration.

To configure embedded header support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. Select the session agent where you want this feature.


```
ACMEPACKET(session-agent)# select
<hostname>:
1: asd          realm=      ip=1.0.0.0
2: SIPSA        realm=      ip=10.10.102.1

selection:2
ACMEPACKET(session-agent)#
```
5. Type **request-uri-headers** followed by the headers and press <Enter>. To configure this parameter for multiple headers, enclose the headers in double quotation marks and separate them with spaces. This completes the configuration of embedded header support.


```
ACMEPACKET(session-agent)# request-uri-headers P-Asserted-Identity
```

You can use the **add** and **delete** commands when you enter the **request-uri-headers** parameter to edit the list of Request URI headers; entering a new list of values without the **add** or **delete** commands overwrites the entire list.

```
ACMEPACKET(session-agent)# request-uri-headers add P-Asserted-Identity
ACMEPACKET(session-agent)# request-uri-headers delete P-Asserted-Identity
```

Static SIP Header and Parameter Manipulation

This section explains the SIP header and parameter manipulation feature, which lets the Net-Net 9200 add, modify, and delete SIP headers and parts of SIP headers called SIP header elements. SIP header elements are the different subparts of the header, such as the header value, header parameter, URI parameter and so on (excluding the header name).

To enable the SIP header and parameter manipulation functionality, you create header manipulation rule sets in which you specify header manipulation rules, as well as optional header element rules that operate on specified header elements. You then apply the header manipulation rule set as inbound or outbound for a session agent or SIP interface.

Header Manipulation Rules

Header manipulation rules operate on the header you specify when you configure the rule. A header manipulation rule can also be configured with a list of element rules, each of which would specify the actions you want performed for a given element of this header.

Header Element Rules

Header element rules perform operations on the elements of a header. Header elements include all subparts of a header; excluding the header name. For example, header value, header parameter, URI parameter, and so on.

About SIP Header and Parameter Manipulation

Using the SIP header manipulation rule set, you can cause the Net-Net 9200 to:

- Delete a header based on header name match.
- Delete a header based on header name match as well as header value match.
- Add a header.
- Modify the elements of a header (by configuring header element rules):
 - Add an element to a header.
For example, add a parameter to a header or add a URI parameter to the URI in a header.
 - Delete an element from a header.
For example, delete a parameter from a header or delete a URI parameter from the URI in a header.
 - Modify an element of a header.
For example, replace a FQDN with an IPv4 address in a header or replace the value of a parameter in the header.
 - Delete a message body part
For example, delete the body part if the Content-Type is “application/ISUP”.

Role in Trunk Group URI Feature

SIP header and parameter manipulation plays a role in the trunk group URI feature. You need to set the new-value parameter to one of the trunk group values when configuring SIP header rules, if using this feature. (In addition you can configure session agents and session agents groups on the Net-Net 9200 to insert trunk group URI parameters in the SIP contact header.

For all trunk group URI support, you must set the appropriate parameters in the SIP header manipulation configuration and in the session agent or session agent group configurations.

For trunk group URI support, the SIP header and parameter manipulation configuration tells the Net-Net 9200 where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

See [Trunk Group URIs \(277\)](#) for more information about trunk group URIs.

Configuring SIP Header and Parameter Manipulation

This section explains how to configure SIP header and parameter manipulation. First you create a SIP header manipulation rule set, then the header manipulation rules and optional header element rules you want that rule set to contain. You then configure a session agent or a SIP interface to use the SIP header and parameter manipulation rule set in the inbound and outbound directions.

Creating SIP Header Manipulation Rule Sets

To configure the SIP header manipulation rule set:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router path.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-manipulation** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. **name**—Enter the name you want to use for this rule set.
5. **header-rules**—Define the header manipulation rules you want to include in this rule set.

- 5a. Type **header-rules** and press <Enter>.

```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```

- 5b. **name**—Enter the name of the header to which this rule applies. (The name you enter here must match a header name.)

This is a case-insensitive string that is compared to the header name for matching. You need to create a rule using the long form of the header name and a rule using the compact form of the header name.

Note: The Request-URI header is identified as “request-uri”.

- 5c. **action**—Enter the action you want applied to the header specified in the name parameter. Valid options are:

Option	Description
add	Add a new header, if that header does not already exist.
delete	Delete the header, if it exists
manipulate	Elements of this header will be manipulated according to the element rules configured. See for details.
none	Default value. No action to be taken.

- 5d. **match-value**—Enter the value to be matched (only an exact match is supported) with a header value. The action you specify is only performed if the header value matches.

- 5e. **msg-type**—Enter the message type to which this header rule applies. Valid options are:

Option	Description
any	Both Requests and Reply messages
request	Request messages only
reply	Reply messages only

- 5f. Type **show** to display the header rule configuration values.

6. **element-rules**—Define the element rules you want to use to be performed on the elements of the header specified by the header rule.

- 6a. Type **element-rules** and press <Enter>.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

- 6b. **name**—Enter the name of the element to which this rule applies.

Note: The **name** parameter usage depends on the element type you enter in step 6. For uri-param, uri-user-param, and header-param it is the parameter name to be added, replaced, or deleted. For all other types, it serves to identify the element rule and any name can be used.

- 6c. **type**—Enter the type of element on which to perform the action. Valid options are:

Option	Description
header-value	Entire value of the header.
header-param	Parameter portion of the header.
uri-user	User portion of the SIP URI.
uri-host	Host portion of the SIP U.I
uri-port	Port number portion of the SIP URI.
uri-param	Parameter included in the SIP URI.
uri-header	Header included in a request constructed from the URI.
uri-user-parameter	User parameter of the SIP URI.

- 6d. **action**—Enter the action you want applied to the element specified in the name parameter, if there is a match value. Valid options are:

Option	Description
add	Add a new element, if it does not already exist
replace	Replace the elements
delete-element	Delete the specified element, if it exists. Based on the match value, if entered in step 6f.
delete-header	Delete the specified header, if it exists.

- 6e. **match-val-type**—Enter the type of value that needs to be matched to the match-field entry for the action to be performed. Valid options are:

Option	Description
IP	Element value in the SIP message must be a valid IP address to be compared to the match-value field entry. If the match-value field is empty, any valid IP address is considered a match. If the element value is not a valid IP address, it is not considered a match.

Option	Description
FQDN	Element value in the SIP message must be a valid FQDN to be compared to the match-value field entry. If the match-value field is empty, any valid FQDN is considered a match. If the element value is not a valid FQDN, it is not considered a match.
ANY	Element value in the SIP message is compared with the match-value field entry. If the match-value field is empty, all values are considered a match.

- 6f. **match-value**—Enter the value you want to match against the element value for an action to be performed.
- 6g. **new-value**—Enter the value for an new element or to replace a value for an existing element. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators.
- Absolute values, with which you can use double quotations marks for clarity. You must escape all double quotes and back slashes that are part of an absolute value, and enclose the absolute value in double quotes.

For example:

`"sip:"+$TRUNK_GROUP+"."+$TRUNK_GROUP_CONTEXT`

- Pre-defined parameters always start with a \$. Valid pre-defined parameters are:

Parameter	Description
\$ORIGINAL	Original value of the element is used.
\$LOCAL_IP	IP address of the SIP interface on which the message was received for inbound manipulation; or sent on for outbound manipulation.
\$LOCAL_PORT	Resolves to the value of the UDP or TCP port on the Net-Net SBC.
\$REMOTE_IP	IP address the message was received from for inbound manipulation; or being sent to for outbound manipulation.
\$REMOTE_PORT	Resolves to the value of the UDP or TCP port on the far end device.
\$REMOTE_VIA_HOST	Host from the top Via header of the message is used.
\$TRUNK_GROUP	Trunk group is used.
\$TRUNK_GROUP_CONTEXT	Trunk group context is used.

- Operators are:

Operator	Description
+	Append the value to the end. For example: <code>"acme"+"packet"</code> generates <code>"acmepacket"</code>
+^	Prepends the value. For example: <code>"acme"+"^"packet"</code> generates <code>"packetacme"</code>

Operator	Description
-	Subtract at the end. For example: "112311"-"11" generates "1123"
^-	Subtract at the beginning. For example: "112311"^- "11" generates "2311"

Examples of entries for the **new-value** field.

```
$ORIGINAL+acme
$ORIGINAL+"my name is john"
$ORIGINAL+"my name is \"john\""
```

```
$ORIGINAL-^781+^617
```

- 6h. Type **show** to display the element rule configuration values.
- 6i. Type **done** to save them.
- 6j. Repeat steps 6b through 6j to create additional rules.
- 6k. Type **exit** to return to the header-rules parameters.
7. **methods**—Enter the SIP method names to which you want to apply this header rule. If entering multiple method names, separate them with commas. For example: INVITE, ACK, BYE.

This field is empty by default. If you leave the method field empty, the header-rule is applied to all methods.
8. Type **exit** to return to the sip-manipulation level.
9. Save and activate your configuration.

Configuring a Session Agent

You can configure a session agent to use the SIP header manipulation rule set.

To configure a session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router path.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **in-manipulationid**—Enter the name of the SIP header manipulation rule set you want to apply to inbound SIP packets.


```
ACMEPACKET(session-agent)# in-manipulationid route-stripper
```
5. **out-manipulationid**—Enter the name of the SIP header manipulation rule set you want to apply to outbound SIP packets.


```
ACMEPACKET(session-agent)# out-manipulationid route-stripper
```
6. Save and activate your configuration.

Configuring a SIP Interface

You can configure a interface to use a SIP header manipulation rule set.

To configure a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router path.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **in-manipulationid**—Enter the name of the SIP header manipulation rule set you want to apply to SIP packets in the ingress direction.


```
ACMEPACKET(sip-interface)# in-manipulationid route-stripper
```
5. **out-manipulationid**—Enter the name of the SIP header manipulation rule set you want to apply to SIP packets in the egress direction.


```
ACMEPACKET(sip-interface)# out-manipulationid route-stripper
```
6. Save and activate your configuration.

SIP HMR (Header Manipulation Rules)

SIP header manipulation has been enhanced so that you can manipulate the headers in SIP messages both statically and dynamically. Available in Net-Net OS Release 4.0 and later, static SIP header manipulation allows you to set up rules in your Net-Net SBC configuration that remove and/or replace designated portions of specified SIP headers. Now you can also set up dynamic header manipulation rules, meaning that the Net-Net SBC has complete control over alterations to the header value. More specifically:

- The Net-Net SBC can search a header for dynamic content or patterns. It can search, for example, for all User parts of a URI that begin with 617 and end with 5555 (e.g., 617...5555).
- The Net-Net SBC can manipulate any part of a pattern match with any part of a SIP header. For example, 617 123 5555 can become 617 231 5555 or 508 123 0000, or any combination of those.

To provide dynamic header manipulation, the Net-Net SBC uses regular expressions to provide a high degree of flexibility for this feature. This allows you to search a specific URI when you do not know that value of the parameter, but want to use the matched parameter value as the header value. It also allows you to preserve matched sections of a pattern, and rebuild the header with these sections in addition to any new parameters you want to add.

How It Works

You can apply header manipulation to session agents, SIP interfaces, and realms. You do so by first setting up header manipulations rules, and then applying them in the configurations where they are needed. Within the header manipulation rules, there are sets of element rules that designate the actions that need to be performed on a given header.

Each header rule and each element rule (HMR) have a set of parameters that you configure to identify the header parts to be manipulated, and in what way the Net-Net SBC is to manipulate them. These parameters are explained in detail in the *ACLI Instructions and Examples* section for this feature, but the parameter that can take regular expression values is **match-value**. This is where you set groupings that you want to store, match against, and manipulate.

Generally, you set a header rule that will store what you want to match, and then you create subsequent rules that operate on this stored value. Because header rules and element rules are applied sequentially, it is key to note that a given rule performs its operations on the results of all the rules that you have entered before it. For example, if you want to delete a portion of a SIP header, you would create Rule 1 that stores the value for the purpose of matching, and then create Rule 2 that would delete the portion of the header you want removed (only if the previous rule matches). This prevents removing data that might be used in the other header rules.

Given that you are using regular expression in this type of configuration, this tightly sequential application of rules means that you must be aware of the results to be yielded from the application of the regular expressions you enter. When you set a regular expression match value for the first rule that you enter, the Net-Net SBC stores the results of that match, and then a second rule might exist that tells the Net-Net SBC to use a new value if it the first rule's match value finds a hit.

Consider the example of the following regular expression entry made for a **match-value** parameter: `'Trunk(.+)'`, which might be set as that match value in the first rule you configure. Given a SIP element rule called `uri-param` and the `param-name` `tgid`, it can yield two values:

- Grouping 0—The entire matching string (`Trunk1`)
- Grouping 1—The grouping (`1`)

In turn, these groupings can be referenced in an element rule by using this syntax:

```
$<header rule name>.$<element rule name>.<value>
```

Additional syntax options that can be used with this feature are:

- `$headerName['[ 'index' ]']`
- `$headerName['[ 'index' ]']['.$index]`
- `$headerName['[ 'index' ]']['.$elementName]`
- `$headerName['[ 'index' ]']['.$elementName']['.$index]`

Guidelines for Header and Element Rules

Header rules and element rules share these guidelines:

- References to groupings that do not exist result in an empty string.
- References to element rule names alone result in a Boolean condition of whether the expression matched or not.
- A maximum of ten matches are allowed for a regular expression. Match 0 (grouping 0) is always the match of the entire matching string; subsequent numbers are the results for other groups that match.

Precedence

The Net-Net SBC applies SIP header rules in the order you have entered them. This guards against the Net-Net SBC removing data that might be used in the other header rules.

In addition to providing that guard, this ordering also provides you with ways to use manipulations strategically. For example, you might want to use two rules if you want to store the values of a regular expression. The first rule would store the value of a matched regular expression, and the second could delete the matched value.

Duplicate Header Names

If more than one header exists for the header name you have configured, the Net-Net SBC stores the value where it can be referenced with the optional syntax `$<header rule name>[index]`. Additional stored header values are indexed in the order in which they appear within the SIP message, and there is no limit to the index.

Possible index values are:

- ~ — The Net-Net SBC references the first matching header
- * — The Net-Net SBC references all headers
- ^ — The Net-Net SBC references the last stored header in the header rule

Performing HMR on a Specific Header

HMR has been enhanced so that you can now operate on a specific instance of a given header. The syntax you use to accomplish this is similar to that you used to refer to a specific header rule stored value instance.

Using the header-name parameter, you can now add a trailing [`<index>`] value after the header name. This [`<index>`] is a numerical value representing the specific instance of the header on which to operate. However, the Net-Net SBC takes no action if the header does not exist. You can also use the caret (^) to reference the last header of that type (if there are multiple instances)

The count for referencing is zero-based, meaning that the first instance of the header counts as 0.

Note that the header instance functionality has no impact on HMR's add action, and you cannot use this feature to insert headers into a specific location. Headers are added to the end of the list, except that Via headers are added to the top.

Specific Header Manipulation Example

Objective:

1. Delete the last "Route" header.

Net-Net SBC HMR Configuration:

sip-manipulation	
name	DelLastHdr
description	
header-rule	
name	DelLastRoute
header-name	Route[[^]]
action	delete
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	
methods	INVITE

Input:

```

INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Route: <sip:9785551111@192.168.1.60:5060;user=phone>
Route: <sip:9785552222@192.168.210.60:5060;user=PHONE>
Route: <sip:9785563333@192.178.120.90:5060>
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 135

```

Output:

```

INVITE sip:service@192.168.200.60:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
CSeq: 1 INVITE
Route: <sip:9785551111@192.168.1.60:5060;user=phone>
Route: <sip:9785552222@192.168.210.60:5060;user=PHONE>
Contact: <sip:sipp@192.168.200.61:5060;user=phone;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

```

Multiple SIP HMR Sets

In general you use SIP HMR by configuring rules and then applying those rules to session agents, realms, or SIP interfaces in the inbound or outbound direction. In addition, the Net-Net SBC has a set method for how certain manipulation rules take precedence over others. For instance, inbound SIP manipulation rules defined in a session agent take precedence over any configured for a realm, and the rules for a realm take precedence over SIP interface manipulation rules.

The multiple SIP HMR feature gives you the ability to:

- Apply multiple inbound and outbound manipulations rules to a SIP message
- Provision the order in which the Net-Net SBC applies manipulation rules

The **action** parameter in the manipulation rules now takes the value **sip-manip**. When you set the parameter to **sip-manip**, you then configure the **new-value** parameter with the name of a SIP manipulation rule that you want to invoke. The values for the **match-value**, **comparison-type**, and **methods** parameters for invoked rule are all supported. This means that the manipulation defined by the rules identified in the **new-value** parameter are carried out when the values for the **match-value**, **comparison-type**, and **methods** parameters are true.

The relationship between manipulation rules and manipulation rule sets is created once you load your configuration, meaning that the order in which you enter them

does not matter. It also means that the Net-Net SBC cannot dynamically perform validation as you enter rules, so you should use the ACLI **verify-config** command to confirm your manipulation rules contain neither invalid nor circular references. Invalid references are those that point to SIP manipulation rules that do not exist, and circular references are those that create endless loops of manipulation rules being carried out over and over. If you load a configuration exhibiting either of these issues, the Net-Net SBC forces the **action** value for the rule to **none** and the rule will not be used.

MIME Body Manipulation Support

Using the SIP HMR feature set, you can manipulate MIME types in SIP message bodies. While you can manipulate the body of SIP messages or a specific content type using other iterations of SIP HMR, this version gives you the power to change the MIME attachment of a specific type within the body by using regular expressions. To achieve this, you use the **find-replace-all** action type, which enables the search for a particular string and the replacement of all matches for that type. Although you use **find-replace-all** to manipulate MIME attachments, it can also be used to achieve other goals in SIP HMR.

How It Works: Find and Replace All

To manipulate a particular portion of the MIME attachment—as is the case for removing a certain attribute within the content type of **application/sdp**—the Net-Net SBC would need to search the content multiple times because:

- SDP can have more than one media line, and
- The SIP message body can contain more than one **application/sdp**.

The **find-replace-all** action type works for SIP header rules and for element rules. You can use it for all manipulation types from the entire header value, to the URI specific parameters, to MIME attachment.

For this action type, it does not matter what you configure the comparison type—which is atypical for actions types, as the comparison type is vital to the others. **Find-replace-all**, however, binds the comparison type to the pattern rule. Thus, the Net-Net SBC treats the match value as a regular expression, and it ignores any configured comparison type value in favor of the pattern rule. This type of action is both a comparison and action: For each regular expression match within the supplied string, the Net-Net SBC substitutes the new value for that match. Yet if you want to replace a certain portion of the regular expression and not the entire matched expression, you need to use a subgroup of expressions and the right syntax to indicate the sub-group replacement index.

Note that using **find-replace-all** might consume more system resources than other HMR types. Therefore this powerful action type should only be used when another type cannot perform the type of manipulation you require.

You can indicate the sub-group replacement syntax by adding the string `[:n:]` to the end of the regular expression—where *n* is a number between 0 and 9. For example, given the following settings:

- `action=find-replace-all`
- `match-value=sip:(user)@host[:1:]`
- `new-value=bob`

You create a new rule to replace only the user portion of the URI that searches for the regular expression and replaces all instances of the user subgroup with the value **bob**.

Taking advantage of the **find-replace-all**'s recursive nature, you can replace all the 0 digits in a telephone number with 1:

- `action=find-replace-all`
- `match-value=0`
- `new-value=1`

So for the user portion of a URI—or for any other string—with a value **1-781-308-4400** would be replaced as **1-781-318-4411**.

If you leave the **new-value** parameter blank for **find-replace-all**, the Net-Net SBC replaces the matched sub-group with an empty string—an equivalent of deleting the sub-group match. You can also replace empty sub-groups, which is like inserting a value within the second sub-group match. For example,

`user()@host.com[[:1:]]` with a configured **new-value** `_bob` yields `user_bob@host.com`.

When you use **find-replace-all**, you cannot use the following **parameter-type** values: **uri-param-name**, **uri-header-name**, and **header-param-name**. These values are unusable because the Net-Net SBC only uses case-sensitive matches for the match-value to find the parameter name within the URI. Since it can only be found by exact match, the Net-Net SBC does not support finding and replacing that parameter.

Escaped Characters

SIP HMR's support for escaped characters allows for searches for values you would be unable to enter yourself. Because they are necessary to MIME manipulation, support for escaped characters now includes:

- `\f`
- `\n`
- `\r`
- `\t`
- `\v`

New Reserved Word

To allow you to search for carriage returns and new lines, the SIP HMR MIME feature also adds support for the reserved word **\$CRLF**. Because you can search for these value and replace them, you also must be able to add them back in when necessary. Configuring **\$CRLF** in the **new-value** parameter always resolves to `/r/n`, which you normally cannot otherwise enter through the ACLI.

About the MIME Value Type

Introduced to modify the MIME attachment, SIP HMR supports a **mime** value for the **type** parameter in the element rules configuration. Like the **status-code** and **reason-phrase** values, you can only use the **mime** type value against a specific header—which in this case, is Content-Type.

When you set the element rule type to **mime**, you must also configure the **parameter-name** with a value. This step is a requirement because it sets the content-type the Net-Net SBC manipulates in a specific part of the MIME attachment. You cannot leave this parameter blank; the Net-Net SBC does not let

you save the configuration if you do. When you use the **store** action on a multi-part MIME attachment that has different attachment types, the Net-Net SBC stores the final instance of the content-type because it does not support storing multiple instances of element rule stored values.

In the event you do not know the specific content-type where the Net-Net SBC will find the **match-value**, you can wildcard the **parameter-name** by setting with the asterisk (*) as a value. You cannot, however, set partial content-types (i.e., **application/***). So configured, the Net-Net SBC loops through the MIME attachment's content types.

You can set the additional **action** types listed in this table with the described result:

Action Type	Description
delete-element	Removes the matched mime-type from the body. If this is the last mime-type within in message body, the Net-Net SBC removes the Content-Type header.
delete-header	Removes all body content and removes the Content-Type header.
replace	Performs a complete replacement of the matched mime-type with the new-value you configure.
find-replace-all	Searches the specifies mime-type's contents and replaces all matching regular expressions with the new-value you configure
store	Stores the final instance of the content-type (if there are multi-part MIME attachments of various attachment types)
add	Not supported

MIME manipulation does not support manipulating headers in the individual MIME attachments. For example, the Net-Net SBC cannot modify the Content-Type given a portion of a message body like this one:

```
--boundary-1
Content-Type: application/sdp

v=0
o=use1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 10000 RTP/AVP 8
a=rtpmap:8 PCMA/8000/1
a=sendrecv
a=ptime:20
a=maxptime:200
```

Back Reference Syntax

You can use back reference syntax in the **new-value** parameter for header and element rules configurations. Denoted by the use of **\$1**, **\$2**, **\$3**, etc. (where the number refers to the regular expression's stored value), you can reference the header and header rule's stored value without having to use the header rule's name. It instead refers to the stored value of "this" rule.

For example, when these settings are in place:

- header-rule=changeHeader

- action=manipulate
- match-value=(.+)([^;])

you can set the **new-value** as sip:\$2 instead of sip:\$changeHeader.\$2.

You can use the back reference syntax for:

- Header rule **actions** **manipulate** and **find-replace-all**
- Element rule **actions** **replace** and **find-replace-all**

Using back reference syntax simplifies your configuration steps because you do not need to create a store rule and then manipulate rule; the manipulate rule itself performs the store action if the **comparison-type** is set to **pattern-rule**.

Notes on the Regular Expression Library

In the regular expression library, the dot (.) character no longer matches new lines or carriage returns. Conversely, the not-dot does match new lines and carriage returns. This change provides a safety mechanism preventing egregious backtracking of the entire SIP message body when there are no matches. Thus, the Net-Net SBC reduces backtracking to a single line within the body. In addition, there is now support for:

Syntax	Description
\s	Whitespace
\S	Non-whitespace
\d	Digits
\D	Non-digits
\R	Any \r, \n, \r\n
\w	Word
\W	Non-word
\A	Beginning of buffer
\Z	End of buffer
\	Any character including newline, in the event that the dot (.) is not

In addition, there is:

Escaped character shortcuts (\w\W\S\s\d\D\R) operating inside brackets [. . .]

Best Practices

This section lists practices that Acme Packet recommends you follow for successful implementation of this feature.

- Define all storage rules first.

This recommendation is made because each subsequent header rule processes against the same SIP message, so each additional header rules works off of the results from the application of the rule that precedes it.

In general, you want to store values from the original SIP header rather than from the iteratively changed versions.

- Implement rules at the element rule rather than the header rule level.
Header rules should only be a container for element rules.

- When you are creating rules to edit a header, add additional element rules to modify a single header rather than try to create multiple header rules each with one element rule. That is, create multiple element rules within a header rule rather than creating multiple header rules.
- Use header rule names that are not in capital letters; there are predefined rules that are used as macros, and they might conflict with a name that uses capital letters; e.g., `$IP_ADDRESS`.

About Regular Expressions

Two of the most fundamental ideas you need to know in order to work with regular expressions and with this feature are:

- Regular expressions are a way of creating strings to match other string values.
- You can use groupings in order to create stored values on which you can then operate.

To learn more about regex, you can visit the following Web site, which has information and tutorials that can help to get you started: <http://www.regular-expressions.info/>.

Many of the characters you can type on your keyboard are literal, ordinary characters—they present their actual value in the pattern. Some characters have special meaning, however, and they instruct the regex function (or engine which interprets the expressions) to treat the characters in designated ways. The following table outlines these “special characters” or “metacharacters.”

Character	Name	Description
.	dot	Matches any one character, including a space; it will match one character, but there must be one character to match. Literally a . (dot) when bracketed ([]), or placed next to a \ (backslash).
*	star/asterisk	Matches one or more preceding character (0, 1, or any number), bracketed carrier class, or group in parentheses. Used for quantification. Typically used with a . (dot) in the format . * to indicate that a match for any character, 0 or more times. Literally an * (asterisk) when bracketed ([]).
+	plus	Matches one or more of the preceding character, bracketed carrier class, or group in parentheses. Used for quantification. Literally a + (plus sign) when bracketed ([]).
	bar/vertical bar/pipe	Matches anything to the left or to the right; the bar separates the alternatives. Both sides are not always tried; if the left does not match, only then is the right attempted. Used for alternation.

Character	Name	Description
{	left brace	Begins an interval range, ended with } (right brace) to match; identifies how many times the previous singles character or group in parentheses must repeat. Interval ranges are entered as minimum and maximums ({minimum, maximum}) where the character/group must appear a minimum of times up to the maximum. You can also use these character to set magnitude, or exactly the number of times a character must appear; you can set this, for example, as the minimum value without the maximum ({minimum, }).
?	question mark	Signifies that the preceding character or group in parentheses is optional; the character or group can appear not at all or one time.
^	caret	Acts as an anchor to represent the beginning of a string.
\$	dollar sign	Acts as an anchor to represent the end of a string.
[	left bracket	Acts as the start of a bracketed character class, ended with the] (right bracket). A character class is a list of character options; one and only one of the characters in the bracketed class must appear for a match. A - (dash) in between two character enclosed by brackets designates a range; for example [a-z] is the character range of the lower case twenty-six letters of the alphabet. Note that the] (right bracket) ends a bracketed character class unless it sits directly next to the [(left bracket) or the ^ (caret); in those two cases, it is the literal character.
(	left parenthesis	Creates a grouping when used with the) (right parenthesis). Groupings have two functions: • They separate pattern strings so that a whole string can have special characters within it as if it were a single character. • They allow the designated pattern to be stored and referenced later (so that other operations can be performed on it).

Multiple SIP HMR Sets

You can now use a sip-manipulation configuration set as an operation to be called from a second sip-manipulation configuration. To do this, you must set the **action** parameter in a header rule to **sip-manip**. Then, enter another sip manipulation name in the **new-value** parameter.

The association between a manipulation rule and the manipulation set happens after all of the sip-manipulations have been loaded. This means that "on-the-fly" validation of sip-manipulation names can not occur as the user is entering the values

in the ACLI. This also means that order does not matter. The administrator can configure their sip-manipulations in any order since the association will be made after all of them have been loaded. If a configuration which contains an invalid reference or loop is loaded, the action for that header rule will be forced to be "none", resulting in a no operation.

Refer to the following example for an application

1. Add a new header to an INVITE
2. Store the user portion of the request URI
3. Remove all Route headers from the message only if the request URI is from a specific user

```

sip-manipulation
  name
  description
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
  deleteRoute
  delete all Route Headers

sip-manipulation
  name
  description
  headers
    header-rule
      name
      header-name
      action
      comparison-type
      match-value
      msg-type
      new-value
      methods
    header-rule
      name
      header-name
      action
      comparison-type
      match-value
      msg-type
      new-value
      methods
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
  addAndDelete
  Add a New header and delete Route

  addHeader
  New
  add
  case-sensitive
  request
  "Some Value"
  INVITE

  storeRURI
  request-uri
  store
  pattern-rule
  request
  INVITE

  storeUser
  uri-user
  store
  any
  pattern-rule
  305.*

```

new-value	
header-rule	
name	deleteHeader
header-name	request-uri
action	sip-manip
comparison-type	Boolean
match-value	\$storeRURI.\$storeUser
msg-type	request
new-value	deleteRoute
methods	INVITE

Boolean logic within Content-Type

As an addition to the existing HMR feature, in Content-Type headers, if the comparison type is set to pattern-rule or boolean, the match-value will be used to compare against the header value of Content-Type, or evaluate the Boolean logic in the match-value. A store operation at either the header-rule or element-rule level stores the header-value and not the body of the message.

ACLI Instructions and Examples

This section explains the parameters that appear in the subelements for the SIP manipulations configuration. Within the SIP manipulations configuration, you can set up SIP header rules, and within those header rules you can configure element rules.

This section also contains several configuration examples for different applications of the HMR feature.

Configuring SIP Header Manipulation Rules

To configure dynamic SIP header manipulation rules:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# header-rules
```
5. **name**—Enter the unique identifier for this SIP HMR. There is no default for this value.
6. **header-name**—Enter the name of the header on which you want the Net-Net SBC to use this HMR. There is no default for this parameter.
7. **msg-type**—Specify the type of message to which this SIP HMR will be applied. There is no default for this parameter.
8. **methods**—Enter the method type to use when this SIP HMR is used, such as **INVITE**, **ACK**, or **CANCEL**. When you do not set the method, the Net-Net SBC applies the rule across all SIP methods.

9. **comparison-type**—Enter the way that you want SIP headers to be compared from one of the available options; the default is **case-sensitive**. This choice dictates how the Net-Net SBC processes the match rules against the SIP header.
 - **case-sensitive**
 - **case-insensitive**
 - **pattern-rule**
 - **refer-case-sensitive**
 - **refer-case-insensitive**
 - **boolean**
10. **action**—Enter the action that you want this rule to perform on the SIP header:
 - **add**
 - **delete**
 - **manipulate**
 - **store**
 - **none**

Remember that you should enter rules with the action type **store** before you enter rules with other types of actions.

When you set the action type to **store**, the Net-Net SBC always treats the match value you enter as a regular expression. As a default, the regular expression is **.+** (which indicates a match value of at least one character), unless you set a more specific regular expression match value.
11. **match-value**—Enter the value to match against the header value in SIP packets; the Net-Net SBC matches these against the entire SIP header value. This is where you can enter values to match using regular expression values.
12. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Net-Net SBC to use when it adds or manipulates SIP headers.
13. Save and activate your configuration.

Configuring SIP Header Manipulation Element Rules

Element rules are a subset of the SIP header manipulation rules and are applied at the element type level rather than at the entire header value.

To configure dynamic SIP header manipulation rules:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# header-rules
```
5. Type **element-rules** and press <Enter>.


```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```

6. **name**—Enter the unique identifier for this element rule. There is no default for this value.
7. **parameter-name**—Enter the SIP header parameter/element on which you want the Net-Net SBC to use this rule.

Do not enter a parameter name if the type you set is `uri-host`, `uri-port`, or `uri-user`.

There is no default for this parameter.
8. **type**—Specify the type of parameter to which this element rule will be applied. There is no default for this parameter, and you can use values such as: `header-value`, `header-param-name`, `header-param`, `uri-display`, `uri-user`, `uri-user-param`, `uri-host`, `uri-port`, `uri-param-name`, `uri-param`, `uri-header-name`, or `uri-header`.
9. **match-val-type**—Enter the value type that you want to match when this rule is applied; you can set this to `IP`, `FQDN`, or `ANY`, and the default is `ANY`.
10. **comparison-type**—Enter the way that you want SIP headers to be compared from one of the available options; the default is `case-sensitive`. This choice dictates how the Net-Net SBC processes the match rules against the SIP header parameter/element.
 - `case-sensitive`
 - `case-insensitive`
 - `pattern-rule`
11. **action**—Enter the action that you want this rule to perform on the SIP header parameter/element:
 - `add`
 - `delete-element`
 - `delete-header`
 - `replace`
 - `store`
 - `none`

Remember that you should enter rules with the action type `store` before you enter rules with other types of actions.

When you set the action type to `store`, the Net-Net SBC always treats the match value you enter as a regular expression. As a default, the regular expression is `.+` (which indicates a match value of at least one character), unless you set a more specific regular expression match value.
12. **match-value**—Enter the value to match against the header value in SIP packets; the Net-Net SBC matches these against the value of the parameter/element. This is where you can enter values to match using regular expression values, or stored pattern matches.
13. **new-value**—When the action parameter is set to `add` or to `manipulate`, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Net-Net SBC to use when it adds or manipulates parameters/elements.
14. Save and activate your configuration.

Testing Pattern Rules

Please test HMR in a controlled environment, such as a lab network before deploying them into the production network. Misconfigured HMRs could adversely affect traffic in unpredictable ways.

Configuration Examples

This section shows you several configuration examples for HMR. This section shows the configuration for the various rules that the Net-Net SBC applied, and sample results of the manipulation.

Example 1: Removing Headers

For this manipulation rule, the Net-Net SBC removes the Custom header if it matches the pattern rule. It stores the defined pattern rule for the goodBye header. Finally, it removes the goodBye header if the pattern rule from above is a match.

This is a sample of the configuration:

```

sip-manipulation
  name
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
  removeHeader
    removeCustom
    Custom
    delete
    pattern-rule
    ^This is my.*
    request
    INVITE
    goodByeHeader
    Goodbye
    store
    pattern-rule
    ^Remove (.+)
    request
    INVITE
    removeHeader
    Goodbye
    delete
    pattern-rule
    $goodByeHeader
    request
    INVITE

```

This is a sample of the result:

```

Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
  Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
  From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
  To: sut <sip:service@192.168.1.61:5060>
  Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
  CSeq: 1 INVITE
  Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
  Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
  Params: sipp <sip:sipp1@192.168.1.60:5060>
  Params: sipp <sip:sipp2@192.168.1.60:5060>
  Edit: disp <sip:user@192.168.1.60:5060>

```

```

Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

```

Example 2: Manipulating the Request URI

For this manipulation rules, the Net-Net SBC stores the URI parameter `tgid` in the Request URI. Then if the pattern rule matches, it adds a new header (`x-customer-profile`) with the a new header value `tgid` to the URI parameter in the request URI.

This is a sample of the configuration:

```

sip-manipulation
  name CustomerTgid
  header-rule
    name
    header-name
    action store
    comparison-type pattern-rule
    match-value
    msg-type request
  new-value
  methods INVITE
  element-rule
    name
    parameter-name tgidParam
    type tgid
    uri-param uri-param
    action store
    match-val-type any
    comparison-type pattern-rule
    match-value
    new-value

  header-rule
    name
    header-name
    action add
    comparison-type pattern-rule
    match-value $ruriRegex.$tgidParam
    msg-type request
    new-value $ruriRegex.$tgidParam.$0
    methods INVITE

  header-rule
    name
    header-name request-uri
    action manipulate
    comparison-type pattern-rule
    match-value $ruriRegex.$tgidParam
    msg-type request
    new-value
    methods INVITE
  element-rule
    name
    parameter-name tgidParam
    type tgid
    uri-param uri-param
    action delete-element
    match-val-type any

```

comparison-type	case-sensitive
match-value	\$ruriRegex.\$tgidParam.\$0
new-value	

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060 SIP/2.0
Message Header
  Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g6plv3088h03acgh6c1.1
  From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rg601-1
  To: sut <sip:service@192.168.1.61:5060>
  Call-ID: SDc1rg601-f125d8b0ec7985c378b04cab9f91cc09-06a3gu0
  CSeq: 1 INVITE
  Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
  Goodbye: Remove Me
  Custom: This is my custom header
  Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
Params: sipp <sip:sipp2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
X-Customer-Profile: 123
```

Example 3: Manipulating a Header

For this manipulation rule, the Net-Net stores the pattern matches for the `Custom` header, and replaces the value of the `Custom` header with a combination of the stored matches and new content.

This is a sample of the configuration:

```
sip-manipulation
  name                               modCustomHdr
  header-rule
    name                             customSearch
    header-name                       Custom
    action                            store
    comparison-type                   pattern-rule
    match-value                       (This is my )(.(+)( header)
msg-type                             request
  new-value
  methods                             INVITE
  header-rule
    name                             customMod
    header-name                       Custom
    action                            manipulate
    comparison-type                   pattern-rule
    match-value                       $customSearch
    msg-type                           request
    new-value
  methods                             INVITE
  element-rule
```

name	hdrVal
parameter-name	hdrVal
type	header-value
action	replace
match-val-type	any
comparison-type	case-sensitive
match-value	
new-value	

\$customSearch.\$1+edited+\$customSearch.\$3

This is a sample of the result:

Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0

Message Header

Via: SIP/2.0/UDP
 192.168.200.61:5060;branch=z9hG4bK20q2s820boghbacgs600.1
 From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDe1ra601-1
 To: sut <sip:service@192.168.1.61:5060>
 Call-ID: SDe1ra601-4bb668e7ec9eeb92c783c78fd5b26586-06a3gu0
 CSeq: 1 INVITE
 Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
 Goodbye: Remove Me
 Custom: This is my edited header
 Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
 Params: sipp <sip:sipp1@192.168.1.60:5060>
 Params: sipp <sip:sipp2@192.168.1.60:5060>
 Edit: disp <sip:user@192.168.1.60:5060>
 Max-Forwards: 69
 Subject: Performance Test
 Content-Type: application/sdp
 Content-Length: 140

header-name	From	
action		store
comparison-type		case-sensitive
match-value		
msg-type		request
new-value		
methods		INVITE
element-rule		
name		elementRule
parameter-name		tag
type		uri-param
action		store
match-val-type		any
comparison-type		case-sensitive
match-value		
new-value		

header-rule		
name		newHeader
header-name		FromTag
action		add
comparison-type		pattern-rule
match-value		\$FromHR.\$elementRule
msg-type		any
new-value		\$FromHR.\$elementRule.\$0

methods

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK4oda2e2050ih7acgh6c1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDf1re601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SDf1re601-f85059e74e1b443499587dd2dee504c2-06a3gu0
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
Params: sipp <sip:sipp2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
FromTag: 1
```

Example 4: Storing and Using URI Parameters

For this manipulation rule, the Net-Net SBC stores the value of the URI parameter `tag` from the `From` header. It also creates a new header `FromTag` with the header value from the stored information resulting from the first rule.

This is a sample of the configuration:

```
sip-manipulation
  name                               storeElemParam
  header-rule
    name                             FromHR
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK4oda2e2050ih7acgh6c1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDf1re601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SDf1re601-f85059e74e1b443499587dd2dee504c2-06a3gu0
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
Params: sipp <sip:sipp2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
```

FromTag: 1

**Example 5:
Manipulating Display
Names**

For this manipulation rule, the Net-Net SBC stores the display name from the Display header. It replaces the two middle characters of the original display name with a new string. Then it also replaces the From header's display name with "abc 123" if it matches "sipp."

This is a sample of the configuration:

```

sip-manipulation
  name
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
      name
      displayName
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
      $storeDisplay.$displayName
      $storeDisplay.$displayName.$1+lur+$storeDisplay.$displayName.$3
  header-rule
    name
    header-name
    action
    comparison-type
    modFrom
    From
    manipulate
    pattern-rule

```

```

match-value
msg-type                request
new-value
methods                INVITE
element-rule
    name                fromDisplay
    parameter-name
    type                uri-display
    action              replace
    match-val-type      any
    comparison-type     pattern-rule
    match-value         sipp
    new-value           "\"abc 123\" "

```

This is a sample of the result:

```

Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681kot109gp04acgs6o0.1
From: "abc 123" <sip:sipp@192.168.1.60:5060>;tag=SD79ra601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SD79ra601-a487f1259e2370d3dbb558c742d3f8c4-06a3gu0
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: slurp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
Params: sipp <sip:sipp2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

```

Example 6: Manipulating Element Parameters

For this more complex manipulation rule, the Net-Net SBC:

- From the Display header, stores the display name, user name, URI parameter up, and header parameter hp
- Adds the header parameter `display` to the Params header, with the stored value of the display name from the first step
- Add the URI parameter `user` to the Params header, with the stored value of the display name from the first step
- If the URI parameter match succeeds in the first step, replaces the URI parameter `up` with the Display header with the value `def`
- If the header parameter match succeeds in the first step, deletes the header parameter `hp` from the Display header

This is a sample of the configuration:

```

sip-manipulation
    name                elemParams
    header-rule
        name            StoreDisplay

```

	header-name	Display
	action	store
	comparison-type	case-sensitive
	match-value	
	msg-type	request
	new-value	
	methods	INVITE
	element-rule	
name	displayName	
	parameter-name	
	type	uri-display
	action	store
	match-val-type	any
	comparison-type	pattern-rule
	match-value	
	new-value	
	element-rule	
	name	userName
	parameter-name	user
	type	uri-user
	action	store
	match-val-type	any
	comparison-type	pattern-rule
	match-value	
	new-value	
element-rule		
	name	uriParam
	parameter-name	up
	type	uri-param
	action	store
	match-val-type	any
	comparison-type	pattern-rule
	match-value	
	new-value	
	element-rule	
	name	headerParam
	parameter-name	hp
	type	header-param
	action	store
	match-val-type	any
	comparison-type	pattern-rule
	match-value	
	new-value	
header-rule		
	name	EditParams
	header-name	Params
	action	manipulate
	comparison-type	case-sensitive
	match-value	
	msg-type	request
	new-value	
	methods	INVITE
	element-rule	
	name	addHeaderParam

	parameter-name	display
	type	header-param
	action	add
match-val-type	any	
	comparison-type	case-sensitive
	match-value	
	new-value	
\$StoreDisplay.\$displayName.\$0		
element-rule		
	name	addUriParam
	parameter-name	user
	type	uri-param
	action	add
	match-val-type	any
	comparison-type	case-sensitive
	match-value	
	new-value	
\$StoreDisplay.\$userName.\$0		
header-rule		
	name	EditDisplay
	header-name	Display
	action	manipulate
	comparison-type	case-sensitive
	match-value	
	msg-type	request
	new-value	
	methods	INVITE
element-rule		
	name	replaceUriParam
	parameter-name	up
	type	uri-param
	action	replace
	match-val-type	any
	comparison-type	pattern-rule
	match-value	\$StoreDisplay.\$uriParam
	new-value	def
element-rule		
	name	delHeaderParam
	parameter-name	hp
	type	header-param
	action	delete-element
	match-val-type	any
	comparison-type	pattern-rule
	match-value	
\$StoreDisplay.\$headerParam		
new-value		

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK7okvei0028jgdacgh6c1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD89rm601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SD89rm601-b5b746cef19d0154cb1f342cb04ec3cb-06a3gu0
```

```

CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sipp <sip:user@192.168.1.60:5060;up=def>
Params: sipp
<sip:sipp1@192.168.1.60:5060;user=user>;display=sipp
Params: sipp <sip:sipp2@192.168.1.60:5060;user=user>;display=sipp
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

```

Example 7: Accessing Data from Multiple Headers of the Same Type

For this manipulation rule, the Net-Net SBC stores the user name from the Params header. It then adds the URI parameter c1 with the value stored from the first Params header. Finally, it adds the URI parameter c2 with the value stored from the second Params header.

This is a sample of the configuration:

```

sip-manipulation
  name
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
  header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
  methods
  element-rule
    name
    parameter-name
    type
    action

```

Params

```

storeParams
Params
store
case-sensitive
request
INVITE
storeUserName
user
uri-user
store
any
case-sensitive
modEdit
Edit
manipulate
pattern-rule
request
INVITE
addParam1
c1
uri-param
add

```

```

                                match-val-type          any
                                comparison-type          case-sensitive
                                match-value
                                new-value
$storeParams[0].$storeUserName.$0
                                element-rule
name                            addParam2
                                parameter-name          c2
                                type                    uri-param
                                action                  add
                                match-val-type          any
                                comparison-type          case-sensitive
                                match-value
                                new-value
$storeParams[1].$storeUserName.$0

```

This is a sample of the result:

```

Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
  Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK9g855p30cos08acgs600.1
  From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD99ri601-1
  To: sut <sip:service@192.168.1.61:5060>
  Call-ID: SD99ri601-6f5691f6461356f607b0737e4039caec-06a3gu0
  CSeq: 1 INVITE
  Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
  Goodbye: Remove Me
  Custom: This is my custom header
  Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
  Params: sipp <sip:sipp1@192.168.1.60:5060>
  Params: sipp <sip:sipp2@192.168.1.60:5060>
  Edit: disp <sip:user@192.168.1.60:5060;c1=sipp1;c2=sipp2>
  Max-Forwards: 69
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: 140

```

Example 8: Using Header Rule Special Characters

For this manipulation rule, the Net-Net SBC:

- Stores the header value of the Params header with the given pattern rule, and stores both the user name of the Params header and the URI parameter `abc`
- Adds the URI parameter `lpu` with the value stored from the previous Params header
- If any of the Params headers match the pattern rule defined in the first step, adds the URI parameter `apu` with the value `apu`
- If all of the Params headers match the pattern rule defined in the first step, adds the URI parameter `apu` with the value `apu`
- If the first Params headers does not match the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter `not` with the value `123`
- If the first Params headers matches the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter `yes` with the value `456`

This is a sample of the configuration:

```

sip-manipulation
  name
  header-rule
    name
    header-name
    Params
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
  header-rule
    name
    add-user-last
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
  $searchParams[^].$userName.$0
  element-rule
    name
    any-param-user
    parameter-name
    type
    action

```

```

                                match-val-type          any
                                comparison-type          pattern-rule
                                match-value             $searchParams[~]
                                new-value              aup
                                element-rule
name                            allParamUser
                                parameter-name          apu
                                type                    header-param
                                action                  add
                                match-val-type          any
                                comparison-type          pattern-rule
                                match-value             $searchParams[*]
                                new-value              apu
                                element-rule
                                name                    notParamYes
                                parameter-name          not
                                type                    uri-param
                                action                  add
                                match-val-type          any
                                comparison-type          pattern-rule
                                match-value             !$searchParams.$emptyUriParam
                                new-value              123
                                element-rule
                                name                    notParamNo
                                parameter-name          yes
                                type                    uri-param
                                action                  add
                                match-val-type          any
                                comparison-type          pattern-rule
                                match-value             $searchParams.$emptyUriParam
                                new-value              456

```

This is a sample of the result:

```

Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
Message Header
  Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681m9t30e0qh6akgj2s1.1
  From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDchrc601-1
  To: sut <sip:service@192.168.1.61:5060>
  Call-ID: SDchrc601-fcf5660a56e2131fd27f12fcbd169fe8-06a3gu0
  CSeq: 1 INVITE
  Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
  Goodbye: Remove Me
  Custom: This is my custom header
  Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
  Params: sipp <sip:sipp1@192.168.1.60:5060>
  Params: sipp <sip:sipp2@192.168.1.60:5060>
  Edit: disp
<sip:user@192.168.1.60:5060;lpu=sipp2;apu=aup;not=123>;apu=apu
  Max-Forwards: 69
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: 140

```

SIP HMR: Response Manipulation

The Net-Net SBC's HMR feature has been enhanced to support the ability to change the status code or reason phrase in SIP responses. This addition—the ability to edit status-lines in responses—builds on HMR's existing ability to edit response headers or the Request-URI in a request.

This section shows you how to configure SIP HMR when you want the Net-Net SBC to drop a **183 Session Progress** response when it does not have SDP, though flexibility is built into this feature so that you can use it to achieve other ends. In addition, you can now set the SIP manipulation's **match-value** parameter with Boolean parameters (**and** or **or**).

ACLI Instructions and Examples

There are new values for configuration parameters and value used for SIP HMR; this section shows you how to configure them.

When you are done with the multi-step process defined in this section, save and activate your configuration.

Setting the Header Name

SIP header rules (part of the SIP manipulation configuration) now support a new value for the header-name parameter. The value is **@status-line**, where the at-sign (@)—not allowed in SIP header names—prevents undesired matches with header having the name "status-code."

To set the header name for SIP header rules:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```
5. **header-name**—Enter the new value for the **header-name** parameter:


```
@status-line.
```

Setting the Element Type

In the element rules (a subset of the SIP header rules configuration), you can now set the **type** parameter to either of the following values, both of which will only have an impact on the status-line:

- **status-code**—Designates the status code of the response line; accepts any string, but during the manipulation process only recognizes the range from 100 to 699
- **reason-phrase**—Designates the reason of the response line; accepts any string

Like other rule types you can set, the Net-Net SBC matches against the value for these using case-sensitive, case-insensitive, or pattern-rule matching (set in the **comparison-type** parameter for the element rule).

To set the element type:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.


```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```
5. Type **element-rule** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```
6. **type**—Enter either **status-code** or **reason-phrase**, the value of which will be used for matching according to the **comparison-type**.

Setting the Match Value

Note that for the SIP header rules and for the SIP element rules, the **match-value** parameter can now be set with these Boolean operators:

- **and** (for which the syntax is the ampersand &)
- **or** (for which the syntax is the pipe |)

However, you can only use Boolean operators in this value when you set the **comparison-type** parameter to **pattern-rule** and are evaluating stored matches. The Net-Net SBC evaluates these Boolean expressions from left to right, and does not support any grouping mechanisms that might change the order of evaluation. For example, the Net-Net SBC evaluates the expression **A & B | C** (where A=true, B=false, and C=true) as follows: **A & B = false; false | true = true**.

You can set the match-value for the SIP header rules or for the SIP element rules.

To set a match value in the SIP header rules configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```

4. Type **header-rules** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```
5. **match-value**—Enter the value to match against the header value in SIP packets; the Net-Net SBC matches these against the entire SIP header value. This is where you can enter values to match using regular expression values; your entries can contain Boolean operators.

To set a match value in the SIP element rules configuration:

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

```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.

```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.

```
ACMEPACKET(session-router)# header-rules
ACMEPACKET(sip-header-rules)#
```
5. Type **element-rule** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)#
```
6. **match-value**—Enter the value to match against the header value in SIP packets; the Net-Net SBC matches these against the value of the parameter/element. This is where you can enter values to match using regular expression values, or stored pattern matches; your entries can contain Boolean operators.

Sample Configuration

Given that the object of this example is to drop the **183 Session Progress** response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the **183 Session Progress** response
2. Determine if the identified **183 Session Progress** responses contain SDP; the Net-Net SBC searches the **183 Session Progress** responses where the content length is zero
3. If the **183 Session Progress** response does not contain SDP, change its status code to 699
4. Drop all 699 responses

```
sip-manipulation
  name                                manip
  description
  header-rule
      name                                IsContentLength0
```


	header-name	Content-Length
	action	store
	comparison-type	pattern-rule
	match-value	0
	msg-type	reply
	new-value	
	methods	
	header-rule	
	name	is183
	header-name	@status-line
	action	store
	comparison-type	pattern-rule
	match-value	
	msg-type	reply
	new-value	
	methods	
	element-rule	
name	is183Code	
	parameter-name	
	type	status-code
	action	store
	match-val-type	any
	comparison-type	pattern-rule
	match-value	183
	new-value	
	header-rule	
	name	change183
	header-name	@status-line
	action	manipulate
	comparison-type	case-sensitive
	match-value	
	msg-type	reply
	new-value	
	methods	
	element-rule	
	name	make699
parameter-name		
	type	status-code
	action	replace
	match-val-type	any
	comparison-type	pattern-rule
	match-value	\$IsContentLength0 & \$is183.\$is183Code
	new-value	699

HMR Functional Enhancements

This section describes several enhancements to the HMR feature. Examples are provided where needed.

Parenthesis Support

HMR now supports using parentheses for both strings and Boolean expressions to support order of operations and to simplify otherwise, complicated manipulation rules.

Boolean example

Now you can evaluate

```
$rule1 & ($rule2 | $rule3)
```

in one manipulation. Prior to this enhancement, the expression above would require two manipulations in order to perform the OR evaluation first.

String example

You can now evaluate

```
(sip: + $rule1.$0) - (user + $rule2.$3)
```

in one manipulation. Prior to this enhancement, the expression above would require three manipulations.

ACLI-based Syntax Checking

When configuring match values or new values, the ACLI presents you with an error message if your syntax is incorrect in either of the following ways.

1. Referencing a non-existent rule or stored value
2. Combining Boolean and string operators within the same expression.

Escape Character Support

Escape characters are supported now in both match-value and new-value parameters. This allows for the escaping of both Boolean and string manipulation operators, the rule identifier character ('\$' which precedes a rule name), parentheses or whitespace to be treated as a regular string. The escape character is a back slash "\".

For example, the string "\+1234" will be treated literally as "+1234"; the "+" will not be a string operator.

Boolean Comparison Type

By setting a comparison-type to **Boolean**, you can evaluate a referenced rule by name to determine if its regular expression or Boolean expression matches to either TRUE or FALSE.

For example, use the Boolean comparison-type for the following match value configurations: "\$Rule1" or "(\$Rule1 & (\$Rule2.\$elem4 | \$Rule3.\$elem1))".

The Boolean comparison type can also be used to determine if a specific match has stored any value. For instance, "\$rule1.\$elem1.\$2" can be used in a match-value. If the stored match "\$2" has any value, the resulting expression will evaluate to TRUE. Conversely, if the specific stored match is empty, the resulting expression will be FALSE. This is particularly useful if a regular expression was used which contains optional groupings.

Note: Previously, the comparison-type, pattern-rule, was overloaded to mean Boolean comparison as well as using a regular expression to compare the match value. This overloading has been decoupled and pattern-rule only means a regular expression comparison. If you still use pattern-rule as a comparison-type correctly, the Net-Net SBC will convert it to a boolean expression.

The boolean value appears in the comparison type parameter in both the sip-header-rules and sip-element-rules configuration elements.

Example Message Headers

Input used in the following examples:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Route: <sip:9785551111@192.168.1.60:5060;user=phone>
Route: <sip:9785552222@192.168.210.60:5060;user=PHONE>
Route: <sip:9785563333@192.178.120.90:5060>
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 135
```

ACLI Instructions and Examples

To configure a boolean comparison type:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# header-rules
```

If you are editing a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
5. **comparison-type**—Enter **boolean** for a boolean comparison.
6. Save your work using the ACLI **done** command.
7. Save and activate your configuration.

Example

Objective:

1. Store uri-user portion of the "Contact" header with a regular expression that contains an optional grouping
2. Change the "Subject" header to include a new header-parameter if the optional sub-grouping from #1 does not exist

Net-Net SBC HMR Configuration:

```
sip-manipulation
  name                               BooleanStoredValue
  description
  header-rule
    name                             StoreContact
    header-name                       Contact
```

```

        action
        comparison-type
        match-value
        msg-type
        new-value
        methods
        element-rule
            name
            parameter-name
            type
            action
            match-val-type
            comparison-type
            match-value
            new-value
header-rule
    name
    header-name
    action
    comparison-type
    match-value
    msg-type
    new-value
    methods
    element-rule
        name
        parameter-name
        type
        action
        match-val-type
        comparison-type
        match-value
        new-value
store
case-sensitive
request
INVITE
StoreUser
uri-user
store
any
pattern-rule
si(I)?pp
ChangeSubject
Subject
manipulate
boolean
!$StoreContact.$StoreUser.$1
request
INVITE
addHP
name
header-param
add
any
case-sensitive
1234

```

Output:

```

INVITE sip:service@192.168.200.60:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
CSeq: 1 INVITE
Route: <sip:9785551111@192.168.1.60:5060;user=phone>
Route: <sip:9785552222@192.168.210.60:5060;user=PHONE>
Route: <sip:9785563333@192.178.120.90:5060>
Contact: <sip:sipp@192.168.200.61:5060;user=phone;transport=udp>
Max-Forwards: 69
Subject: Performance Test;name=1234
Content-Type: application/sdp
Content-Length: 140

```

Header Instance Selection

You can specify the specific instance of a header to act upon. The header-name field can now accept a trailing [*<index>*] after the header name. Index is a numerical value

representing the specific instance of the header to act upon. Referencing headers is zero based, meaning the 1st header of any type starts at count 0, and not 1. If a header index is referenced for a header which does not exist, no operation is taken.

Header Instance Shortcuts

The ASCII character '^' can also be used to reference the last header of that type (if multiple instances exist) and the '*' character can be used to act on all headers of that type. The '*' behavior is the existing behavior, and therefore the default behavior.

Note: If you configure `contact[*]` as the ACLI header-name, a subsequent listing of the header rule would only appear as `contact` and not `contact[*]`, as the '*' is superfluous.

This header instance functionality has no effect on the "add" HMR action. It can not be used to insert headers into a specific location. To effectively insert a header into a specific location, you must perform three header manipulation steps:

1. Store all headers
2. Delete all headers
3. Reconstruct header appearance from stored values.

Headers are always added at the end of the list, unless they are Vias which get added to the top.

Example

Objective:

1. Delete the last "Route" header.

Net-Net SBC HMR Configuration:

sip-manipulation		
name		DelLastHdr
description		
header-rule		
name		DelLastRoute
header-name		Route[^]
action		delete
comparison-type		case-sensitive
match-value		
msg-type		request
new-value		
methods		INVITE

Output:

```
INVITE sip:service@192.168.200.60:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
CSeq: 1 INVITE
Route: <sip:9785551111@192.168.1.60:5060;user=phone>
Route: <sip:9785552222@192.168.210.60:5060;user=PHONE>
Contact: <sip:sipp@192.168.200.61:5060;user=phone;transport=udp>
Max-Forwards: 69
Subject: Performance Test
```

Content-Type: application/sdp
Content-Length: 140

Stored Value Case Discrimination

The new comparison types **refer-case-sensitive** and **refer-case-insensitive** are used when you want to compare the *stored* value, or pattern-rule sub-group against a header parameter value, URI parameter value, or header value by either a case-sensitive, or case-insensitive compare, respectively. Previously you could only compare a statically configured value.

Generally, this comparison type is used when a combination of 2 or more header or element rules are used to comprise a full SIP manipulation.

The **refer-case-sensitive** and **refer-case-insensitive** values appear in the comparison type parameter in both the sip-header-rules and sip-element-rules configuration elements.

ACLI Instructions and Examples

To configure a stored value comparison type with or without case discrimination:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **header-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# header-rules
```

If you are editing a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
5. **comparison-type**—Enter **refer-case-sensitive** or **refer-case-insensitive** depending on your application.
6. Save your work using the ACLI **done** command.
7. Save and activate your configuration.

Dialog-Matching Header Manipulation

The most common headers to manipulate using HMR are the To-URI and From-URI. Along with the to-tag, from-tag, and Call-ID values, these are also all headers that represent dialog-specific information that must match the UAC and UAS to be considered part of the same dialog. If these parameters are modified through HMR, the results can be that the UAC or UAS rejects messages.

While it is possible to ensure that dialog parameters match correctly using regular HMR, this feature offers a simpler and less error-prone method of doing so.

In addition, this section describes the addition of “built-in” SIP manipulations defined by Acme Packet best practices, and a new method of testing your SIP manipulations.

About Dialog-Matching Header Manipulations

The goal of this feature is to maintain proper dialog-matching through manipulation of dialog-specific information using HMR. Two fundamental challenges arise when looking at the issue of correctly parameters manipulating dialog-matching:

- Inbound HMR
- Outbound HMR

The new setting **out-of-dialog** (for the **msg-type** parameter) addresses these challenges by offering an intelligent mod of dialog matching of messages for inbound and outbound HMR requests. This is a **msg-type** parameter, meaning that it becomes matching criteria for operations performed against a message. If you also specify methods (such as REGISTER) as matching criteria, then the rule is further limited to the designated method.

For both inbound and outbound manipulations, using the **out-of-dialog** setting means the message must be a request without a to-tag in order to perform the manipulation.

Inbound HMR Challenge

Since inbound manipulations take place before the message reaches the core of Net-Net SBC SIP processing, the SIP proxy takes the manipulated header as what was directly received from the client. This can cause problems for requests leaving the Net-Net SBC for the UAC because the dialog will not match the initial request sent.

So the unmodified header must be cached because for any subsequent request (as in the case of a BYE originating from the terminator; see the diagram below) the Net-Net SBC might need to restore the original value—enabling the UAC to identify the message correctly as being part of the same dialog. For out-of-dialog requests (when the To, From, or Call-ID headers are modified) the original header will be stored in the dialog when the **msg-type out-of-dialog** is used.

The Net-Net SBC performs the restoration of original headers outside of SIP manipulations. That is, there are no manipulation rules to configure for restoring the header to their original context. The Net-Net SBC will recognize the headers that have been modified, and restore them to their original state prior to sending the message out on the wire. Restoration takes place prior to outbound manipulations so that any outbound manipulation can those headers once they have been restored.



Outbound HMR Challenge

When you use the **out-of-dialog** setting for an outbound manipulation, the Net-Net SBC only executes this specific SIP header rule only if the same SIP header rule was executed against the initial dialog-creating request.

For example, if the INVITE's To header was not manipulated, it would not be correct to manipulate the To header in the BYE request. To do so would render the UAC unable to properly match the dialog. And this also means that the outbound manipulation should be carried out against a To, From, or Call-ID header in the BYE request if it was manipulated in the INVITE.



ACLI Instructions and Examples

You using the **out-of-dialog** setting in the **msg-type** parameter, part of the SIP header rules configuration.

To enable dialog-matching header manipulation:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **mime-rules** and press <Enter>. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.


```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
```
5. **msg-type**—Set this parameter to **out-of-dialog** to enable dialog-matching header manipulation. See the [Inbound HMR Challenge \(231\)](#) and [Outbound HMR Challenge \(232\)](#) sections for more information about how this setting works.
6. Save your work.

Built-In SIP Manipulations

In the course of HMR use, certain rules have become commonly used. Lengthy and complex, these rules do not include any customer-specific information and so they can be used widely. To make using them easier, they have been turned into built-in rules that you can reference in the **in-manipulationid** and **out-manipulationid** parameters that are part of the realm, session agent, and SIP interfaces configurations.

Built-in rules start with the prefix **ACME_**, so Acme Packet recommends you name your own rules in a different manner to avoid conflict.

While the number of built-in manipulation rules is expected to grow, one is supported at the present time: **ACME_NAT_TO_FROM_IP**. When performed outbound, this rule changes:

- The To-URI hostname to the logical **\$TARGET_IP** and port to **\$TARGET_PORT**
- The From-URI to the logical **\$REPLY_IP** and port to be **\$REPLY_PORT**

ACLI Instructions and Examples

When you want to enable this feature for a realm, session agent, or SIP interface, you configure the **in-manipulationid** or **out-manipulationid** parameters with the rule.

The sample here shows this feature being applied to a session agent, but the realm and SIP interface configurations also have the same parameter you use to set up the feature.

To use built-in SIP manipulations:

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


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

2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **out-manipulationid**—Enter name of the built-in rule you want to use. Remember that all built-in rules start with **ACME_NAT_TO_FROM_IP**.
5. Save your work.

Testing SIP Manipulations

You can now use a new tool that allows you to test the outcome of your SIP manipulation and header rules without sending real traffic through the Net-Net SBC.

To use the tool, you enter the ACLI's test-sip-manipulation utility and reference the rule you want to test using its name. Then you enter a mode where you put in a SIP message entered in ASCII. You can cut and paste this message from `sipmsg.log` or from some other location. Using <Ctrl-D> stops the SIP message collection and parses it.

The test informs you of any parsing errors found in the SIP message. Once the message is entered, you can execute the SIP manipulation against the message. The output after this step is the modified SIP message after manipulations have been applied. You will also find a debugging option, which displays SIP manipulation logging to the screen as the manipulation takes place.

As a starting point for testing, this tool comes loaded with a default SIP message. It cannot be associated with realms, session agents, or SIP interfaces, and so it also comes with certain resolves reserved words, such as: `$LOCAL_IP`, `$TRUNK_GROUP_CONTEXT`, and `$REMOTE_PORT`. In addition, you can use your settings for testing across terminal sessions; if you choose to save your settings, everything (including the SIP message) will be saved, with the exception of the debugging option.

It is not recommended that you use this tool to add an ISUP message body.

Unique HMR Regex Patterns

In addition to the HMR support it offers, the Net-Net SBC can now be provisioned with unique regex patterns for each logical remote entity. This supplement to pre-existing HMR functionality saves you provisioning time and saves Net-Net SBC resources in instances when it was previously necessary to define a unique SIP manipulation per PBX for a small number of customer-specific rules.

Manipulation Pattern Per Remote Entity

On the Net-Net SBC, you can configure logical remote entities (session agents, realms, and SIP interfaces) with a manipulation pattern string that the system uses as a regular expression. Then the SIP manipulation references this regular expression using the reserved word `$MANIP_PATTERN`. At runtime, the Net-Net SBC looks for the logical entity configured with a manipulation pattern string in this order of preference: session agent, realm, and finally SIP interface.

On finding the logical entity configured with the manipulation string, the Net-Net SBC dynamically determines the expression. When there is an invalid reference to a

manipulation pattern, the pattern-rule expression that results will turn out to be the default expression (which is \, +).

When the `$MANIP_PATTERN` is used in a manipulation rule's **new-value** parameter, it resolves to an empty string, equivalent of no value. Even though this process ends with no value, it still consumes system resources. And so Acme Packet recommends you do not use `$MANIP_PATTERN` as a **new-value** value.

In the following example, the SIP manipulation references the regular expression from a realm configuration:

```

realm-config
  identifier                net200
  description
  addr-prefix               0.0.0.0
  network-interfaces       public:0
  ...
  manipulation-pattern      Lorem(.+)
sip-manipulation
  name                      manip
  description
  header-rules
    name                    headerRule
    header-name             Subject
    action                  manipulate
    match-value             $MANIP_PATTERN
    msg-type                request
    comparison-type         pattern-rule
    new-value               Math
    methods                 INVITE

```

Reject Action

Release D7.0.0 adds a new action type called **reject** to all manipulation rules. When you use this action type and a condition matching the manipulation rule arises, the Net-Net SBC rejects the request (though does not drop responses) and increments a counter.

- If the **msg-type** parameter is set to **any** and the message is a response, the Net-Net SBC increments a counter to show the intention to reject the message—but the message will continue to be processed.
- If the **msg-type** parameter is set to **any** and the message is a request, the Net-Net SBC performs the rejection and increments the counter.

The **new-value** parameter is designed to supply the status code and reason phrase corresponding to the reject. You can use the following syntax to supply this information: **status-code[:reason-phrase]**. You do not have to supply the status code and reason phrase information; by default, the system uses **400:Bad Request**.

If you do supply this information, then the status code must be a positive integer between 300 and 699. The Net-Net SBC then provides the reason phrase corresponding to the status code. And if there is no reason phrase, the system uses the one for the applicable reason class.

You can also customize a reason phrase. To do so, you enter the status code followed by a colon (:), being sure to enclose the entire entry in quotation marks (") if your reason code includes spaces.

When the Net-Net SBC performs the **reject** action, the current SIP manipulation stops processing and does not act on any of the rules following the **reject** rule. This course of action is true for nested SIP manipulations that might have been constructed using the **sip-manip** action type.

ACLI Instructions and Examples

To support the **reject** action, two parameters in the **session-router-config** allow you to set how many messages in a certain amount of time cause the Net-Net SBC to generate an SNMP trap.

To set the reject message number and time window:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-router** and press <Enter>.


```
ACMEPACKET(session-router)# session-router
ACMEPACKET(session-router-config)#
```
4. **reject-message-threshold**—Enter the minimum number of message rejections allowed in the **reject-message-window** time on the Net-Net SBC (when using the SIP manipulation action reject) before generating an SNMP trap. The default is 0, meaning this feature is disabled and no trap will be sent.
5. **reject-message-window**—Enter the time in seconds that defines the window for maximum message rejections allowed before generating an SNMP trap. The default is 0, meaning this feature is disabled and no trap will be sent.
6. Save your work.

About Counters

The Net-Net SBC tracks messages that have been flagged for rejection using the **reject** action type. In the **show sip transport** display, refer to the Rejected Messages category; there is no distinction between requests and responses.

```
ACMEPACKET# show sip transport
19:23:27-199 sip1s@0.4.1
SIP Transport
```

	---- Recent ----			----- Lifetime -----		
	Active	High	Total	Total	PerMax	High
Server Trans	0	2	4	19	3	3
Client Trans	0	1	2	17	3	2
Context IDs	0	2	4	19	3	3
Sockets	0	0	0	0	0	0
Inbound Connections	0	0	0	0	0	0
Outbound Connections	0	0	0	0	0	0
Connection Aliases	0	0	0	0	0	0
Pending Messages	0	0	0	0	0	0
Connections Rejected	-	-	0	0	0	
Connection Errors	-	-	0	0	0	
Pending Msg Dropped	-	-	0	0	0	
Reg Cache Updates	-	-	0	0	0	
Overload Rejects	-	-	0	0	0	

Suppressed Retransmit	-	-	0	0	0	
Response Retransmit	-	-	20	50	10	
Request Dropped	-	-	0	0	0	
Response Dropped	-	-	0	0	0	
Invalid Requests	-	-	0	0	0	
Invalid Responses	-	-	0	0	0	
Invalid Messages	-	-	0	0	0	
Transaction Timeouts	-	-	0	0	0	
Duplicate Transactions	-	-	0	0	0	
Transaction Errors	-	-	0	0	0	
Application Errors	-	-	0	0	0	
Redundancy Errors	-	-	0	0	0	
Rejected Message(10s)	0	1	2	5	1	1

SNMP Support

The net-Net SBC provides SNMP support for the Rejected Messages data, so you can access this information externally. The new MIB objects are:

```

apSysRejectedMessages    OBJECT-TYPE
    SYNTAX                Counter32
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        "Number of messages rejected by the SD due to matching
        criteria."
    ::= { apSysMgmtMIBGeneralObjects 18 }

apSysMgmtRejectedMessagesThresholdExeededTrap    NOTIFICATION-TYPE
    OBJECTS { apSysRejectedMessages }
    STATUS                 current
    DESCRIPTION
        " The trap will be generated when the number of rejected messages
        exceed the configured threshold within the configured window."
    ::= { apSystemManagementMonitors 57 }

apSysMgmtRejectedMessagesGroup    OBJECT-GROUP
    OBJECTS {
        apSysRejectedMessages
    }
    STATUS                 current
    DESCRIPTION
        "Objects to track the number of messages rejected by
        the SD."
    ::= { apSystemManagementGroups 18 }

apSysMgmtRejectedMessagesNotificationsGroup    NOTIFICATION-GROUP
    NOTIFICATIONS {
        apSysMgmtRejectedMesagesThresholdExeededTrap
    }
    STATUS                 current
    DESCRIPTION
        "Traps used for notification of rejected messages"
    ::= { apSystemManagementNotificationsGroups 26 }

apSmsgmtRejectedMessagesCap
    AGENT-CAPABILITIES

```

```

PRODUCT-RELEASE      "Acme Packet SD"
STATUS               current
DESCRIPTION          "Acme Packet Agent Capability for
enterprise           system management MIB."
SUPPORTS             APSYSMGMT-MIB
INCLUDES             {
                    apSysMgmtRejectedMessagesGroup,
                    apSysMgmtRejectedMessagesNotificationsGroup
                    }
                    ::= { apSgmtMibCapabilities 37 }

```

Log Action

Release D7.0.0 adds a new action type called **log** to all manipulation rules. When you use this action type and a condition matching the manipulation rule arises, the Net-Net SBC logs information about the current message to a separate log file. This log files will be located on the same core in which the SIP manipulation occurred. On the core where **sip** runs, a logfile called **matched.log** will appear when this action type is executed.

The **matched.log** file contains a timestamp, received and sent Net-Net SBC network interface, sent or received IP address:port information, and the peer IP address:port information. It also specifies the rule that triggered the log action in this syntax: **rule-type[rule:name]**. The request URI, Contact header, To Header, and From header are also present.

```

-----
Apr 17 14:17:54.526 On [0:0]192.168.1.84:5060 sent to 192.168.1.60:5060
element-rule[checkRURIPort]
INVITE sip:service@192.168.1.84:5060 SIP/2.0
From: sipp <sip:+2125551212@192.168.1.60:5060>;tag=3035SIPpTag001
To: sut <sip:service@192.168.1.84>
Contact: sip:sipp@192.168.1.60:5060

```

Changes to Storing Pattern Rule Values

Release D7.0.0 introduces changes to the framework for storing regular expression results within manipulation rules, altering the way the **store** action works. These changes are beneficial to performance.

In previous releases, when the **store** action is used, the Net-Net SBC stores all values matching the regular expression defined in the **match-value** parameter for all headers. At runtime, the system evaluates all stored values to find the correct index.

Now, you no longer need to specify the **store** action. The simple fact of referencing another rule tells the system it must store a value. When SIP manipulation is used, the system first checks to see if any values require storing. The **add** action is an exception to this process; storing happens after a header is added.

When referring to a rule, that rule still needs to have a regular expression defined in the **match-value** and the comparison type set to **pattern-rule**; else the default expression will be used.

Removal of Restrictions

The following restrictions related to HMR have been removed in Release D7.0.0:

- The action **find-replace-all** now executes all element rules. Previously, no child rules were executed.
- The action **sip-manip** now executes existing all element rules. Previously, no child rules were executed.
- The action **store** now executes existing all element rules. Previously, only child rules with the **store** action were executed.
- The action **add** now executes existing all element rules. Previously, only child rules with the **add** action were executed.

Name Restrictions for Manipulation Rules

Historically, you have been allowed to configure any value for the name parameter within a manipulation rule. This method of naming caused confusion when referencing rules, so now manipulation rules name must follow a specific syntax. They must match the expression `^[a-zA-Z0-9_]+$` and contain at least one lower case letter.

In other words, the name must:

- Start with a letter, and then it can contain any number of letters, numbers, or underscores
- Contain at least one lower case letter

All pre-existing configurations will continue to function normally. If you want to change a manipulation rule, however, you are required to change its name if it does not follow the new format.

The ACLI **verify-config** command warns you if the system has loaded a configuration containing illegal naming syntax.

New Value Restrictions

To simplify configuration and remove possible ambiguity, the use of boolean and equality operators (`==`, `<=`, `<`, etc.) for **new-value** parameter values has been banned. Since there was no specific functionality tied to their use, their ceasing to be use will have no impact to normal SIP manipulation operations.

HMR For SIP-ISUP

See [HMR for SIP-ISUP \(344\)](#) for more information.

Dialog Transparency

This section explains how to configure dialog transparency, which prevents the Net-Net 9200 from generating a unique Call-ID and modifying dialog tags.

Overview

With dialog transparency enabled, the Net-Net 9200 is prevented from generating a unique Call-ID and from modifying the dialog tags; the Net-Net 9200 passes what it receives. Therefore, when a call made on one Net-Net 9200 is transferred to another UA and crosses a second Net-Net 9200, the second Net-Net 9200 does not note the context of the original dialog, and the original call identifiers are preserved end to end. The signalling presented to each endpoint remains in the appropriate

context regardless of how many times a call crosses through a Net-Net 9200 or how many Net-Net SDs a call crosses.

Without dialog transparency enabled, the Net-Net 9200's SIP B2BUA rewrites the Call-ID header and inserted dialog cookies into the From and To tags of all messages it processes. These dialog cookies are in the following format: SDxxxxNN-. Using these cookies, the Net-Net 9200 can recognize the direction of a dialog. However, this behavior makes call transfers problematic because one Net-Net SDs' Call-ID might not be properly decoded by another Net-Net 9200. The result is asymmetric header manipulation and failed call transfers.

ACLI Instructions and Examples

To enable SIP dialog transparency:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
```
4. Use the ACLI **select** command so that you can work with the SIP configuration.


```
ACMEPACKET(sip-config)# select
```
5. **dialog-transparency**—Enter the state of SIP dialog transparency you require for your Net-Net 9200.

SIP Via Transparency

The inbound Via header transparency feature enables the Net-Net 9200 to insert its Via header on top of the top-most Via header received from user equipment (UE). It then forwards it on to the IP Multimedia Subsystem (IMS) core with the original Via header now located as the bottom-most Via header.

The Net-Net 9200 still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.

This feature is targeted for the Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAAN) with SIP hosted NAT traversal support. It works with SIP NAT bridged, local-policy routed, and non-SIP NAT configurations, regardless of registration handling.

How it Works

Some equipment acts as Proxy-CSCF (P-CSCF) and Serving-CSCF (S-CSCF) nodes, with the Net-Net 9200 is located between the equipment and user endpoints. The equipment needs to see the each user endpoint's original Via header in order to perform some implicit authentication, admission, and control functions in a TISPAAN-compliant model.

You enable Via header transparency on the access SIP interface or a session agent. Received Via headers are saved for inclusion in requests going out another interface or session agent that does not have the parameter set, in other words, the core side. For any received SIP message where the inbound previous hop interface was enabled for Via header transparency, the Net-Net 9200 adds its own Via header as it forwards it, and it also copies the received top-most Via as the new bottom-most Via, if the outbound next hop interface/session agent is not enabled for Via header

transparency. The Net-Net 9200 also adds a received= parameter to the copied Via header, per the SIP RFC 3261.

Any message received from an interface without Via header transparency enabled, does not have the received Via header copied over to any other direction.

For HNT, where the original top-most (and only) Via header from a UE is a private/false address, the 9200 should still copy that false address into the core-side, and the received= parameter will contain the real Layer-3 addressing.

ACLI Instructions and Examples

You can configure SIP Via header transparency for the access SIP interface using the ACLI.

To configure SIP Via header transparency for an access interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. You can either add support to a new SIP interface configuration or to an existing SIP interface configuration:
 - 4a. For a new SIP interface configuration, you can add the option by typing **options**, a <Space>, and then **via-header-transparency**.


```
ACMEPACKET(sip-interface)# options via-header-transparency
```
 - 4b. For an existing SIP interface configuration without options configured, select the SIP interface, type **options** followed by a <Space>, and then **via-header-transparency**.


```
ACMEPACKET(sip-interface)# select
ACMEPACKET(sip-interface)# options via-header-transparency
```
 - 4c. For an existing SIP interface configuration with options configured, select the SIP interface, type **options** followed by a <Space>, the plus sign (+), and the **via-header-transparency** option.


```
ACMEPACKET(sip-interface)# select
ACMEPACKET(sip-interface)# options +via-header-transparency
```
5. Save and activate your configuration.

Full Via Header Replacement

In the basic behavior of the SIP VIA Header Transparency feature, the Net-Net SBC retains the top-most VIA header received on ingress and places it under the VIA header containing the system's address on egress. All VIA headers beneath the top-most header in the ingress message are lost.

The Net-Net SBC can also retain all VIA headers received on the message's ingress. Thus the top-most VIA header becomes the second VIA header on egress, the second VIA header becomes the third VIA header on egress, and so on.

ACLI Instructions and Examples

To configure SIP Via header transparency for all received VIA headers:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI **select** command) the SIP interface that you want to edit.
5. For an existing SIP interface configuration without options configured, select the SIP interface, type **options** followed by a <Space>, and then **via-header-transparency=all**.


```
ACMEPACKET(sip-interface)# options via-header-transparency=all
```
6. Save and activate your configuration.

Symmetric Latching

Symmetric latching, or forced HNT, ensures that symmetric RTP/RTCP is used for a SIP endpoint. Symmetric RTP/RTCP means that the IP address and port pair used by an outbound RTP/RTCP flow is reused for the inbound flow. The IP address and port are learned when the initial RTP/RTCP flow is received by the Net-Net 9200. The flow's source address and port are latched onto and used as the destination for the RTP/RTCP sourced by the other side of the call. The IP address and port in the c line and m line respectively in the SDP message are ignored.

If your network is configured with nested realms in order to separate signalling from media, make sure that the symmetric latching feature is enabled on the signalling realm.

Note: This description is applicable to RTCP only when you also enable the HNT RTCP option in the **media-manager** configuration. Do not enable symmetric latching on core-facing interfaces.

ACLI Instructions and Examples

This section contains instructions for how to configure symmetric latching and RTCP symmetric latching.

Configuring Symmetric Latching

To configure symmetric latching:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

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

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. Select the realm where you want to apply this feature.

```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>          0.0.0.0
2: PRIV_Realm <none>          0.0.0.0
3: PBLCREALM <none>          0.0.0.0
```

```
selection:1
ACMEPACKET(realm-config)#
```

5. Type **symmetric-latching enabled** and press <Enter>.

```
ACMEPACKET(realm-config)# symmetric-latching enabled
```

6. Save and activate your configuration.

Configuring RTCP Symmetric Latching

To enable RTCP symmetric latching in the ACLI, you must also take the following steps:

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

```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press <Enter> to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. Select the media manager configuration so that you can enable HNT TRCP.

```
ACMEPACKET(media-manager-config)# select
```

5. Type **hnt-rtcp enabled** and press <Enter>.

```
ACMEPACKET(media-manager-config)# hnt-rtcp enabled
```

6. Save and activate your configuration.

SIP Number Normalization

This section explains the SIP number normalization feature that applies to the SIP To URI. Number normalization includes add, delete, and replace string functions that result in consistent number formats.

How it Works

Number normalization applies to the SIP To URI. It occurs on ingress traffic, prior to the generation of accounting records or local policy lookups. RADIUS CDR attributes are populated with the normalized numbers. Local policy matching is based on the normalized numbers.

Terminology

The following terminology is used in the descriptions contained in the next sections.

- X is any digit having the value 0 through 9
- N is any digit having the value 2 through 9
- 0/1 is a digit having the value of either 0 or 1
- NXX is a form of Numbering Plan Area (NPA).
- CC is a 1, 2, or 3 digit country code used in international dialing
- NN is a national number that can be a four to fourteen digit national number used in international dialing, where the combination of CC+NN is a 7 to 15 digit number.
- + symbol in E.164 indicates that an international prefix is required
- E.164 numbers are globally unique, language independent identifiers for resources on Public Telecommunication Networks that can support many different services and protocols.
- N11 number is any of the three-digit dialing codes in the form N11 used to connect users to special services, where N is a digit between 2 and 9

Calls from IP Endpoints

The Net-Net 9200 uses the following number normalization rules:

- North American Numbering Plan (NANP) calls: where a number with the format **1NPANXXXXXX** is received, the Net-Net 9200 adds a plus sign (+) as a prefix to the NANP number. The Net-Net 9200 also adds the string **;user=phone** after the host IP address in the SIP URI. For example:
`sip:+1NPANXXXXXX@ipaddr;user=phone`
- International NWZ1 calls: Net-Net 9200 receives an international call with the format **011CCNN**. The Net-Net 9200 deletes the **011** prefix and adds a plus sign (+) as a prefix to CC+NN; and also adds the string **;user=phone** after the host IP address in the SIP URI. For example:
`sip:+CCNN@ipaddr;user=phone`
- Private number calls: when a private number with the format **nxxxx** (where n=2 through 9) is received, no number normalization is applied by the Net-Net 9200.
- Calls to numbers such as N11, 0-, 0+, 00-, and 01+: the Net-Net 9200 adds **;phone-context=+1** after the number and also adds the string **;user=phone** after the host IP address in the SIP URI. For example:
`sip:N11;phone-context=+1@ipaddr;user=phone`
`sip:01CCNN;phone-context=+1@ipaddr;user=phone`
- Calls with numbers that are already normalized are not modified by the Net-Net 9200.

Calls from IP Peer Network

For calls received from external peer networks, the Net-Net 9200 uses the following number normalization rules:

- Global numbers such as NANP and international E.164 numbers should have already been normalized. If not, the Net-Net 9200 applies the same number normalization rules listed in the prior section.
- Calls to numbers such as N11, 0-, 0+, 00-, and 01+: the Net-Net 9200 adds ;phone-context=+1 after the number and also adds the string ;user=phone (if absent) after the host IP address in the SIP URI.

ACLI Instructions and Examples

You can configure SIP number normalization for the realm and session agent using the ACLI.

Configuring a Realm

To configure SIP number normalization for a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. You can either add SIP number normalization support to a new session agent configuration or to an existing session agent configuration:
 - For a new realm configuration, add the option by typing **options**, a <Space>, and then **number-normalization**.


```
ACMEPACKET(realm-config)# options number-normalization
```
 - For an existing realm configuration without any options already configured, select the realm, type **options** followed by a <Space>, and then **number-normalization**.


```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)# options number-normalization
```
 - For an existing realm configuration with other options, select the realm, type **options** followed by a <Space>, the plus sign (+), and the **number-normalization** option.


```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)# options +number-normalization
```
5. Save and activate your configuration.

Configuring a Session Agent

To configure SIP number normalization for a session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. You can either add SIP number normalization support to a new session agent configuration or to an existing session agent configuration:

- For a new a session agent configuration, add the option by typing **options**, a <Space>, and then **number-normalization**.

```
ACMEPACKET(session-agent)# options number-normalization
```

- For an existing session agent configuration without any options already configured, select the session agent, type **options** followed by a <Space>, and then **number-normalization**.

```
ACMEPACKET(session-agent)# select
ACMEPACKET(session-agent)# options number-normalization
```

- For an existing session agent configuration with other options, select the session agent, type **options** followed by a <Space>, the plus sign (+), and the **number-normalization** option.

```
ACMEPACKET(session-agent)# select
ACMEPACKET(session-agent)# options +number-normalization
```

5. Save and activate your configuration.

SIP Port Mapping

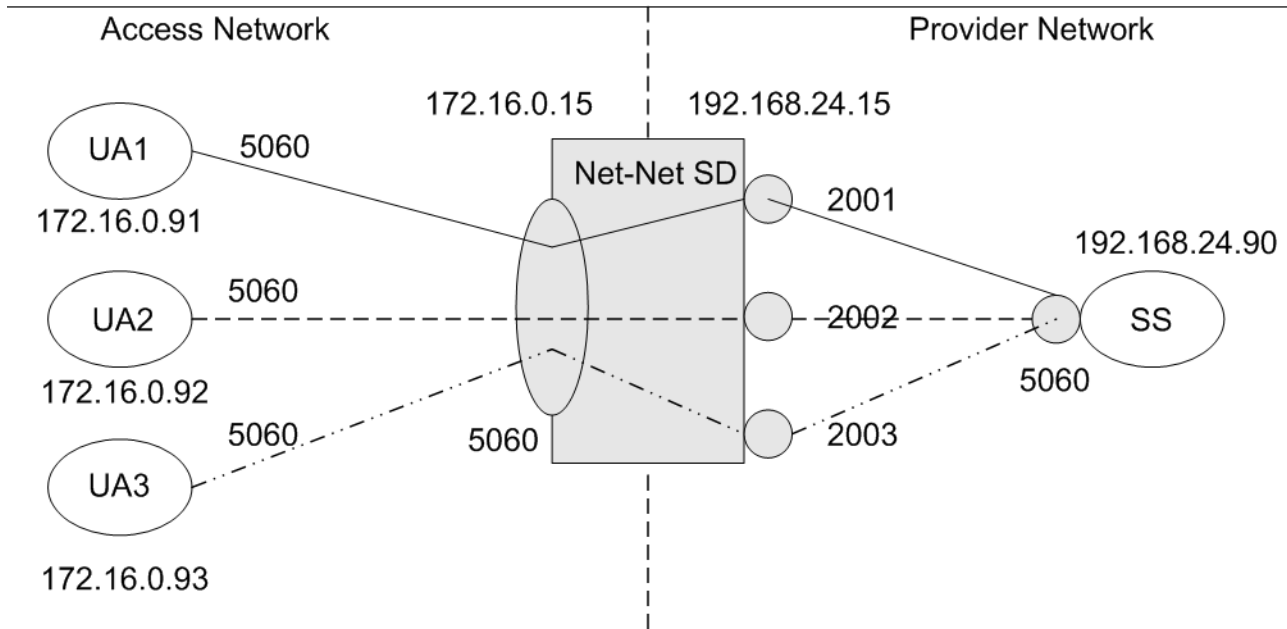
This section contains information about the SIP port mapping feature. SIP port mapping lets you allocate a unique SIP signaling transport address (IP address and UDP port) on the Net-Net 9200 in the provider network for each registered endpoint (user agent).

About SIP Port Mapping

You might need to provide a unique signaling transport address for each registered endpoint for admission control; if required by your softswitch vendor. If you have questions about your softswitch, contact the vendor for assistance.

When a Net-Net 9200 resides between the endpoints and the softswitch, the softswitch sees the same transport address (that of the Net-Net 9200) for all endpoints. By allocating a unique UDP port for each endpoint, the Net-Net 9200 provides each of them a unique transport address.

The following example illustrates the SIP port mapping feature.



The diagram shows UA1, UA2, and UA3 are endpoints within the access network and that the SIP interface for the access network is 172.16.0.15:5060. On the provider network, the SIP interface is at 192.168.24.15, with the SIP port mapping feature enabled. The softswitch/registrars is also located on the provider network at 192.168.24.90:5060.

The diagram shows that port 2001 on the provider network is allocated to UA1 on the access network, port 2002 is allocated to UA2, and port 2003 is allocated to UA3. Because of this allocation, all SIP signaling messages sent from the endpoints in the access network to the softswitch on the provider network travel through an allocated signaling port. For example, all signaling messages between UA1 and the softswitch use 192.168.24.15:2001 as the transport address.

How SIP Port Mapping Works

The Net-Net 9200 allocates SIP port mapping (signaling) ports during a REGISTER request that has registration caching applied. When you define a range of signaling ports for the SIP interface, you create a pool of signaling ports that can be allocated during the REGISTER request.

The Net-Net 9200 allocates a signaling port from the pool when it creates the registration cache entry for a Contact in a REGISTER request. It allocates a separate signaling port for each unique Contact URI from the access side. The registration cache Contact entry contains the mapping between the Contact URI in the access/endpoint realm (the UA-Contact) and the Contact URI in the registrar/softswitch realm (the SD-Contact).

The SD-Contact is the allocated signaling port. The signaling port gets returned to the pool when the Contact is removed from the registration cache. The removal can occur when the cache entry expires; or when the endpoint sends a REGISTER request to explicitly remove the Contact from the registrar. When a signaling port returns to the pool it gets placed at the end of pool list; in a least-recently-used allocation method for signaling ports.

When the Net-Net 9200 forwards the REGISTER request to the softswitch, it replaces the UA-Contact with SD-Contact. For example, if UA1 sends a REGISTER request with a Contact URI of `sip:ua1@172.16.0.91:5060`, it is replaced with `sip:192.168.24.15:2001` when the REGISTER request is forwarded to the registrar.

The same translation occurs when UA1 sends that same URI in the Contact header of other SIP messages. SIP requests addressed to the allocated signaling transport address (SD-Contact) are translated and forwarded to the registered endpoint contact address (UA-Contact).

Note: The maximum number of registered endpoints cannot exceed the number of signaling ports available. If no signaling ports are available for a new registration, the REGISTER request receives a 503 response.

The Net-Net 9200 still processes requests received on the configured SIP port address. Requests sent into the registrar/softswitch realm that are not associated with a registered user will use the configured SIP port address.

About NAT Table ACL Entries

To enable SIP signaling messages to reach the host processor, the Net-Net 9200 adds NAT table ACL entries for each SIP interface. With UDP without SIP port mapping applied, it adds a single ACL entry for each SIP port in the SIP interface configuration. For example:

```
untrusted entries:
intf:vlan source-ip/mask:port/mask dest-ip/mask:port/mask  prot type  index
0/0:0      0.0.0.0                172.16.1.15:5060    UDP  static  10
0/3:0      0.0.0.0                192.168.24.15:5060  UDP  static  16
0/1:0      0.0.0.0                192.168.50.25:5060  UDP  static  17
```

Using SIP Port Mapping

When you use SIP port mapping, one or more ACL entries are added to the NAT table to enable the range of ports defined. The NAT table does not support the specification of port ranges. However, it does support masking the port to enable ranges that fall on bit boundaries. For example, an entry for `192.168.24.15:4096/12` defines the port range of 4096 through 8191.

The algorithm for determining the set of ACLs for the port map range balances the need to represent the range as closely as possible, with the need to minimize the number of ACL entries. For example, a range of 30000 through 39999 would result in the following set of ACLs.

```
untrusted entries:
intf:vlan source-ip/mask:port/mask dest-ip/mask:port/mask  prot type  index
0/3:0      0.0.0.0                192.168.24.15:30000/4  UDP  static  13
0/3:0      0.0.0.0                192.168.24.15:32768/4  UDP  static  14
0/3:0      0.0.0.0                192.168.24.15:36864/4  UDP  static  15
```

However, the first entry actually enables ports 28672 through 32767 and the last entry allows port 36864 through 40959. If SIP messages are received on ports outside the configured range (28672 through 29999 or 40000 through 40959 in this case), they are ignored.

Acme Packet recommends you use port map ranges that fall on bit boundaries to ensure the fewest possible ACL entries are created and only the configured ports are allowed by the ACLs. For example, a range of 32768 to 49151 provides for 16,384 signaling ports in a single ACL entry (`192.168.24.15:32768/2`).

Note: If the ACLs added for the port map range do not include the SIP port configured in the SIP interface; the normal SIP ACL entry for the SIP port is also added.

Dynamic Configuration

Dynamic configuration of SIP port mapping can cause disruption in service for existing registration cache entries; depending on the changes made to the defined port map range. If the range of mapping ports is reduced, it is possible that SIP signaling messages from the registrar/softswitch realm will no longer be sent to the host processor because of the changes in the NAT Table ACL entries.

When the range of mapping ports is changed, any signaling ports in the free signaling port pool not allocated to a registration cache entry are removed from the pool. When an allocated signaling port that is no longer part of the defined mapping port range is released, it is not returned to the pool of free steering ports.

The administrator is warned when the changed configuration is activated after the port map range of a SIP interface has been changed.

Configuring SIP Port Mapping

You configure the SIP port mapping feature on a per-realm basis using the SIP interface configuration. Configure the port map range on the SIP interface for the realm where the registrar/softswitch resides. Port mapping is only applied when the access/ingress realm has registration caching and/or HNT enabled.

The range of SIP mapping ports must not overlap the following:

- Configured SIP port, which might be used for signaling messages not associated with a registered endpoint.
- Port range defined for steering pool configuration using the same IP address as the SIP interface. If overlap occurs, the NAT table entry for the steering port used in a call prevents SIP messages from reaching the host processor.

ACLI Instructions and Examples

To configure SIP port mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router path.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **port-map-start**—Set the starting port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. If you set the value to zero (0), SIP port mapping is disabled.


```
ACMEPACKET(sip-interface)# port-map-start 32768
```
5. **port-map-end**—Set the ending port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. If you set the value to zero (0), SIP port mapping is disabled.

Note: If not set to zero (0), the ending port must be greater than the starting port.

```
ACMEPACKET(sip-interface)# port-map-end 40959
```

6. Save and activate your configuration, but note the caution in the [Dynamic Configuration \(249\)](#) section just above.

The following example shows the above SIP port mapping configured parameters for a SIP interface:

```
sip-interface
    port-map-start          32768
    port-map-end            40959
```

SIP Registration Overload Protection

You can configure your Net-Net SBC for SIP Registration overload protection, which augments the Net-Net SBC's pre-existing protection methods. Working with the Net-Net SBC's access control and registration caching functions, this new feature guards against benign avalanche restarts. The avalanche is caused by events where many endpoints lose power or connectivity at once, are restored to service, and then flood the Net-Net SBC as they attempt to register again.

How It Works

Normally, the Net-Net SBC handles SIP registration by creating a temporary registration cache for the endpoint's address of record (AoR) and forward the REGISTER request to the registrar. To challenge the endpoint's registration, the registrar sends back either a **401 Unauthorized** or **407 Proxy Authorization Required** response. When it receives the 401 or 407, the Net-Net SBC saves the challenge context in anticipation of receiving a second REGISTER with the endpoint's authentication credentials. The Net-Net SBC forwards the second REGISTER (with authentication credentials) to the registrar, and then the registrar confirms registration with a 200 OK. Both REGISTER requests are subject to the Net-Net SBC's access control rules, set either for the ingress realm or the ingress session agent. The Net-Net SBC also honors the maximum registration sustain rate constraint for session agents; this applies when the incoming REGISTER is from a session agent and the outgoing REGISTER is sent to a session agent.

When you enable SIP Registration overload protection, the Net-Net SBC temporarily promotes the endpoint to the trusted level when it receives the 401 or 407 response (to the first REGISTER) from the registrar. This ensures that the second REGISTER (containing authentication credentials) can reach the Net-Net SBC. Temporary promotion lasts only for the amount of time remaining before the REGISTER server transaction expires plus the time allotted in the transaction expiration parameter in the SIP configuration. Before the temporary promotion expires, there is enough time for any necessary retransmission of the first REGISTER and for the second REGISTER to take place. The following situations might also occur:

- If the Net-Net SBC receives a 401 or 407 to the second REGISTER request, it resets its access control level for the endpoint's address to the default level; it then treats additional REGISTER requests from the same context at the default access control level.
- If the Net-Net SBC receives a 200 OK response to the REGISTER message, it extends the promotion time to the expiration period for the registration cache.

If the Net-Net SBC is unable to find the temporary registration cache and the saved challenge context when the second REGISTER arrives, it forwards the REGISTER without checking the maximum registration sustain rate constraint for ingress and egress session agents—thereby ensuring that the REGISTER with authentication credentials is sent to the registrar. So when you use this feature, you should set the maximum registration sustain rate constraint of the session agent (representing the registrar) at half the registrar's maximum registration sustain rate. Additional REGISTER requests with the same challenge context are subject to the maximum registration sustain rate constraint.

ACLI Instructions and Examples

When you configure this feature, be sure to set the following two options in your global SIP configuration:

- reg-overload-protect
- cache-challenges

To enable SIP Registration overload protection on your Net-Net SBC:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**reg-overload-protect**), and then press <Enter>. Follow the same steps to add the **cache-challenges** option.


```
ACMEPACKET(sip-config)# options +reg-overload-protect
ACMEPACKET(sip-config)# options +cache-challenges
```

If you type either of these options without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.
5. Save and activate your configuration.

SIP Request Method Throttling

You can configure throttling mechanisms for SIP INVITEs and REGISTERs using session agent constraints. However, you might want to throttle other types of SIP methods, and for those methods you should use the rate constraints configuration available both in the session constraints (which you then apply to a SIP interface or a realm) and the session agent configurations.

Acme Packet recommends you use session agent constraints for session-rate INVITE throttling and registration-rate for REGISTER throttling.

For SIP access deployments, you can configure rate constraints for individual method types along with a set of burst and sustain rates. These constraints can help to avoid overloading the core network. In addition, they restrain the load non-INVITE messages use, thus reserving capacity for INVITE-based sessions and Registrations

How It Works

When you configure SIP request method throttling, you must exercise care because it is possible to reject in-dialog requests. Therefore, Acme Packet recommends you do NOT configure constraints—although the configuration allows you to and will not produce error messages or warnings if you set them—for the following SIP method types:

- ACK
- PRACK
- BYE

- INFO
- REFER

However, the Net-Net SBC is likely to throttle NOTIFY requests despite their being part of a Subscribe dialog.

Therefore, the methods you will most likely configure for throttling are:

- NOTIFY
- OPTIONS
- MESSAGE
- PUBLISH
- REGISTER

The Net-Net SBC counts Re-INVITEs and challenged responses against the throttle limit, but does not check to determine if the constraints have been exceeded for either.

You can configure separate constraints—inbound and outbound values for burst and sustain rates—for each different method type you configure. Although you should use session agent constraints (and not rate constraints) for INVITEs, if you also set up rate constraints for INVITEs, then the smallest configured value takes precedence.

About Counters and Statistics

Each rate constraint you configure for a SIP method tracks its own counters. For example, if you configure a rate constraint for the PUBLISH method, the burst and sustain rates you set for it apply only to the PUBLISH method and not to any other methods for which you might set up rate constraints. You can, however, set the burst rate window in the session constraints configuration that will apply to all methods configured as rate constraints.

The Net-Net SBC captures statistics for SIP methods throttled by rate constraints for SIP interfaces and session agents; it does not capture these statistics for the global SIP configuration.

ACLI Instructions and Examples

This section shows you how to set up rate constraints for session constraints (which are then applied to SIP interfaces) and session agents.

Requirements

To use this feature, you must enable the **extra-method-stats** parameter in the global SIP configuration.

To set the extra-method-stats parameter in the global SIP configuration:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-config** and press <Enter>.

```
ACMEPACKET(session-router)# sip-config
```

```
ACMEPACKET(sip-config)#
```

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

4. **extra-method-stats**—Set this parameter to **enabled**.
5. Save and activate your configuration.

Rate Constraints for SIP Interfaces

To apply rate constraints to SIP interfaces, you need to configure rate constraints in the session constraints configuration and then apply the session constraints to the SIP interface where you want them used.

Note that you need to set up the parent **session-constraint** configuration to save any rate constraints you configure.

To configure rate constraints:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **session-constraints** and press <Enter>.

```
ACMEPACKET(session-router)# session-constraints
ACMEPACKET(session-constraints)#
```

If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. Type **rate-constraints** and press <Enter>.

```
ACMEPACKET(session-constraints)# rate-constraints
ACMEPACKET(rate-constraints)#
```

5. **method**—Enter the SIP method name for the method you want to throttle. Although the parameter accepts other values, your entries should come only from the following list for the feature to function properly:

- NOTIFY
- OPTIONS
- MESSAGE
- PUBLISH
- REGISTER

6. **max-inbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
7. **max-outbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
8. **max-inbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound sustain rate on the SIP

interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.

9. **max-outbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
10. Save your changes and apply this session constraint and its rate constraint(s) to SIP interfaces.

Applying Session and Rate Constraints to a SIP Interface

You need the name of the session constraints configuration to apply the restrictions you set up to a SIP interface.

To apply session and rate constraints to a SIP interface:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **session-constraints**—Enter the name of the session constraint configuration where you have set up rate constraints to apply them to this SIP interface. This parameter has no default, and must be the valid name of a session constraint configuration.
5. Save and activate your configuration.

Configuring Rate Constraints for Session Agents

You can also use this feature for individual SIP session agents.

To configure rate constraints for a SIP session agent:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **session-agent** and press <Enter>.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. Type **rate-constraints** and press <Enter>.

```
ACMEPACKET(session-agent)# rate-constraints
ACMEPACKET(rate-constraints)#
```

5. **method**—Enter the SIP method name for the method you want to throttle. Your entries should come only from the following list:
 - NOTIFY
 - OPTIONS
 - MESSAGE
 - PUBLISH
 - REGISTER
6. **max-inbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
7. **max-outbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
8. **max-inbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
9. **max-outbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
10. Save and activate your configuration.

SIP Delayed Media Update

The Net-Net SBC supports SIP delayed media update. When enabled, this feature keeps the Net-Net SBC from updating its media flow information for flows established after an offer-answer exchange. The Net-Net SBC does not update the flow information until a new offer and answer arrive for a specific set of media flows. The (subsequent) offer does not have to be for the same session; rather, it can appear as a new SIP INVITE that uses the same SDP.

Delayed Media Update Disabled

When this feature is disabled (the default behavior), the Net-Net SBC updates media flow entries in its CAM based on signaled SDP when it processes the SDP. If it processes an SDP offer, the Net-Net SBC allocates steering port resources; the Net-Net SBC updates any missing elements for the flow when the answer is returned.

In cases when a secondary offer arrives (either a reINVITE, an UPDATE, or the original INVITE is hairpinned back through the Net-Net SBC), the Net-Net SBC updates the following media flow information at the time of the offer:

- Destination IP address
- Destination port
- Realm for the media flows
- Media release settings

This behavior affects specific applications that are better served by the Net-Net SBC waiting to update media flow information until it receives the answer to the second offer.

Delayed Media Update Enabled

When you enable the SIP delayed media update feature, the Net-Net SBC:

- Delays changing the active media flow CAM entry for a new offer if a previous offer and answer have been received for the same media flows; it encodes new SDP information in an outgoing offer, but does not change the CAM entry until the answer is received
- Delays changing the active media flow CAM entry even when the new offer is for a new session
- Supports media release when performing delayed media update changes
- Offers per-realm configuration

How It Works

This section describes how the delayed media update feature works for hairpinned call flows and for an SDP offer arriving for installed flows.

- **Hairpinned call flows**—In this type of call flow, the application server (AS) sends an INVITE back to the Net-Net SBC and that INVITE needs to be forwarded to another user (user B). When it receives the offer in this INVITE and delayed media update is disabled, the Net-Net SBC determines that the call is hairpinned and deletes the CAM entry for the flow for user A, who has sent the initial INVITE. The Net-Net SBC deletes the CAM entry for the flow from the AS to user A.

With delayed media update enabled, the CAM entry for the flow from the AS to user A is not deleted. Instead, the Net-Net SBC waits until it has an answer from user B, and then performs the necessary updates and deletions.

- **SDP offer for installed media flows**—With delayed media update enabled, if it has received an offer and answer and a new offer arrives for the same flow, the Net-Net SBC delays updating the CAM entries until an answer is received for the new offer.

ACLI Instructions and Examples

You enable this feature on a per-realm basis by setting one parameter.

Tenable SIP delayed media update:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The prompt changes to let you know you can configure individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **delay-media-update**—Change this parameter from disabled (default) to enabled.
5. Save and activate your configuration.

SIPConnect

The Net-Net SBC supports the SIPconnect model, wherein PBXs register themselves so that service providers do not need to know IP addresses or locations in advance for static configurations. This is particularly helpful when the PBX is behind a NAT.

In the PBX registration process, the PBX creates a binding between one of its phone numbers as the address of record (AoR) and Contact-URI in the REGISTER message. The registrar knows that the single AoR actually represents many addresses, and so it registers them implicitly. However, the registrar does not return the implicit AoR number in P-Associated-URIs.

The SIPconnect feature resolves the following issues that arise from using this model:

- SIP INVITEs sent to the PBX from the Registrar through the Net-Net SBC have the Request-URI of registered contact. Because it typically ignores the To-URI, the PBX needs the Request-URI username portion to be the specific extension number being called.

With the SIP connect feature enabled, the Net-Net SBC overwrites the Request-URI username with the To-URI username.

- SIP INVITEs from the PBX have the From AoR and Contact-URI usernames of specific phones rather than of the registered AoR and Contact-URI. For the Net-Net SBC, this means that it cannot use the **allow-anonymous** parameter value of **register**; there would be no registered user matches, and the Net-Net SBC would reject them (with a 403 Forbidden).

With the SIP connect feature enabled, the Net-Net SBC performs allow-anonymous checking based on the registered Via address, which is the same for all requests for the same PBX.

Modifications to Registration Caching Behavior

With the SIP connect feature enabled, Net-Net SBC registration caching works the same way that it does with the feature disabled, with the following exceptions:

The Net-Net SBC determines whether the destination realm has the **sip-connect-pbx-reg** option configured, and then:

- If it is configured, the Net-Net SBC replaces the user part of the Request-URI with the user part of the To header. When the INVITE contains a P-Called-Party-ID header, the Net-Net SBC uses the user part of the P-Called-Party-ID header (instead of the To header).
- If it is not configured, the Net-Net SBC determines if the destination address is for a session agent and whether that session agent has **sip-connect-pbx-reg** option configured. When it is configured, the Net-Net performs the same replacements described in the bullet directly above. When it is not configured, the Net-Net SBC does not make any replacements.

When it receives an INVITE request, the Net-Net SBC checks the incoming realm for the **sip-connect-pbx-reg** option.

- If it is configured, the Net-Net SBC uses the INVITE's source address (instead of the AoR and Contact-URI) to search the registration cache for a matched registration entry.
- If it is not configured, the Net-Net SBC determines if the INVITE's source address is for a session agent and whether that session agent has **sip-connect-pbx-reg** option configured.

When it is configured, the Net-Net SBC replaces the user part of the Request-URI with the user part of the To header. When the INVITE contains a P-Called-Party-ID header, the Net-Net SBC uses the user part of the P-Called-Party-ID header (instead of the To header).

When it is not configured, the Net-Net SBC does not make any replacements.

Configuring SIP Connect Support

You configure this feature by adding the `sip-connect-pbx-reg` option to the realm configuration. In addition, though this feature requires that your configuration also be set up as outlined in this section. The first two items are required, and Acme Packet recommends that you also implement the suggested additional configuration.

Required Configuration

- Registration caching is enabled.
- For the realm from which registrations come, the options list must include `sip-connect-pbx-reg`; this is new configuration introduced to support this feature. The presence of this option instructs the Net-Net SBC to skip matching the Contact header in the INVITE request with the registered Contact of the registration entry. The Net-Net SBC finds a registration using only the INVITE's source address.

Alternatively, you can configure the `sip-connect-pbx-reg` option in the options list for a session agent. When the realm where an INVITE comes from does not have this option set, the Net-Net SBC determines whether or not the INVITE came from a session agent. You might choose to configure session agents with this option if you do not want it applied to an entire realm. If the PBX is behind a NAT device, the session agent's IP address for the PBX (if statically configured) must be the IP address of the NAT device. And if DNS is use, the session agent's hostname must resolve to the NAT device's IP address.

Suggested Additional Configuration

- In the SIP ports configuration (accessed through the SIP interface configuration), the `allow-anonymous` parameter must be set to `registered`. This setting allows the Net-Net SBC to accept SIP requests from session agents and registered endpoints only, but to accept REGISTER requests from any endpoint.
- For the SIP interface that accepts registrations, the `options` parameter must be set to `reg-via-key`. This setting allows the Net-Net SBC to use the source address of an INVITE as the key to find a registration entry in the registration cache. When the INVITE's Contact header matches the registered Contact in the registration entry, the Net-Net SBC accepts the INVITE request.

ACLI Instructions and Examples

To set the SIP connect option for a realm configuration:

1. Check the [Configuring SIP Connect Support \(259\)](#) section of this document for configuration prerequisites and notes.
1. In Superuser mode, type `configure terminal` and press <Enter>.


```
ACMEPACKET# configure terminal
```
2. Type `media-manager` and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type `realm-config` and press <Enter>.

ACMEPACKET(media-manager)# **realm-config**

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **sip-connect-pbx-reg** with a “plus” sign in front of it, and then press <Enter>.

ACMEPACKET(realm-config)# **options +sip-connect-pbx-reg**

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

To set the SIP connect option for a SIP session agent configuration:

1. Check the [Configuring SIP Connect Support \(259\)](#) section of this document for configuration prerequisites and notes.
2. In Superuser mode, type **configure terminal** and press <Enter>.

ACMEPACKET# **configure terminal**

3. Type **session-router** and press <Enter> to access the signaling-related configurations.

ACMEPACKET(configure)# **session-router**

4. Type **session-agent** and press <Enter>.

ACMEPACKET(session-router)# **session-agent**

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.

5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **sip-connect-pbx-reg** with a “plus” sign in front of it, and then press <Enter>.

ACMEPACKET(session-agent)# **options +sip-connect-pbx-reg**

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the session agent’s configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

SIP Registration Burst Rate Control

The Net-Net SBC offers you a new way to control SIP registration burst rates to smooth the registration rate.

This feature adds to the Net-Net SBC’s pre-existing ability to limit the:

- Number of new register requests sent to a session agent (using the max-register-sustain-rate parameter)
- Number of forwarded registrations from already-registered endpoints (using the max-forwards option)

While these methods successfully guard against the Net-Net SBC's becoming overwhelmed with register requests, registration rates can still be too bursty and more registrations can occur within a short time than a registrar can handle.

SIP registration burst rate control allows you to configure two parameters per SIP session agent—one that controls the registration burst rate to limit the number of new registration requests, and a second to set the time window for that burst rate. When the registration rate exceeds the burst rate you set, the Net-Net SBC responds to new registration requests with **503 Service Unavailable** messages.

Note that this constraint is not applied to re-registers resulting from a **401 Unauthorized** challenge request.

ACLI Instructions and Examples

To configure SIP registration burst rate control:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.
4. **register-burst-window**—Define the window size in seconds for the maximum number of allowable SIP registrations. 0 is the minimum and default value for this parameter; the maximum value is 999999999.
5. **max-register-burst-rate**—Enter the maximum number of new registrations you want this session agent to accept within the registration burst rate window. 0 is the minimum and default value for this parameter; the maximum value is 999999999.
6. Save your work using the ACLI **done** command.
7. Save and activate your configuration.

SIP Mid-Call Re-Registration Following Service Route

In prior releases, the **route-register-no-service-route [all | refresh]** option was made available in the **sip-config** element, in order to control how a User Agent (UA) is re-registered. The following are definitions of the option and its arguments:

- **route-register-no-service-route**—This option prevents the use of the Service-Route procedure to route the Re-Register requests after the UA has initially registered.
- **route-register-no-service-route=all**—Prevents the use of the Service-Route procedure to route the Re-Register requests for all messages, after the UA has initially registered.
- **route-register-no-service-route=refresh**—Prevents the use of the Service-Route procedure to route the Re-Register requests for all refresh-

register messages, but not de-register messages, after the UA has initially registered.

For some users, however, depending on their network, re-registering UAs while a call is taking place can cause problems and allow the call to be torn down.

How It Works

A new `idle` value has been developed for the `route-register-no-service-route` to ensure that, when enabled, the Net-Net SBC follows the previously defined rules for idle calls, where `idle` means not engaged in any Invite-based sessions.

ACLI Instructions and Examples

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
```
4. **options**—Set the options parameter by entering **options**, a <Space>, the option name, either **route-register-no-service-route=all;idle** or **route-register-no-service-route=refresh;idle**, with a “plus” sign in front of it, and then press <Enter>. For example:


```
ACMEPACKET(sip-config)# options +route-register-no-service-route=all;idle
```

If you enter an option without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the **sip-config**’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

SIP Session Agent Group Recursion

Expanding on the Net-Net SBC's current routing capabilities, this feature gives you the ability to configure a SIP session agent group (SAG) to try all of its session agents rather than to the next-best local policy match if the first session agent in the SAG fails.

In prior releases, the Net-Net SBC performs routing by using local policies, trunk group URIs, cached services routes, and local route tables. Local policies and trunk group URIs can use SAGs to find the most appropriate next-hop session agent based on the load balancing scheme you choose for that SAG: round robin, hunt, proportional distribution, least busy, and lowest sustained rate. When it locates a SAG and selects a specific session agent, the Net-Net SBC tries only that single session agent. Instead of trying other members of the SAG, the Net-Net SBC recurses to the local policy that is the next best match. This happens because the Net-Net SBC typically chooses a SAG based on the fact that it has not breached its constraints, but the Net-Net SBC only detects failed call attempts (due to unreachable next hops, unresolved ENUM queries, or SIP 4xx/5xx/6xx failure responses) after it has checked constraints. So the Net-Net SBC only re-routes if there are additional matching local policies.

When you enable SIP SAG recursion, the Net-Net SBC will try the additional session agents in the selected SAG if the previous session agent fails. You can also set specific response codes in the SAG configuration that terminate the recursion. This method of terminating recursion is similar to the Net-Net SBC's ability to stop recursion for SIP interfaces and session agents.

How It Works

You enable SIP SAG recursion on a per-SAG basis by setting new parameters in the SAG (**session-group**) configuration.

Session agents are selected according to the strategy you set for the SAG, and these affect the way that the Net-Net SBC selects session agents when this feature enabled:

- Round robin and hunt—The Net-Net SBC selects the first session agent according to the strategy, and it selects subsequent session agents based on the order they are entered into the configuration.
- Proportional distribution, least busy, and lowest sustained rate—The Net-Net SBC selects session agents based on the list of session agents sorted by the criteria specified.

You can terminate recursion based on SIP response codes that you enter into the SAG configuration. You can configure a SAG with any SIP response code in the 3xx, 4xx, and 5xx groups. Since you can also set such a list in the session agent configuration, this list is additive to that one so that you can define additional codes for a session agent group without having to repeat the ones set for a session agent.

ACLI Instructions and Examples

To enable SIP SAG recursion and set response code to terminate recursion:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **session-group** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-group
```

```
ACMEPACKET(session-group)#
```

If you are adding support for this feature to a pre-existing SAG, then you must select (using the CLI **select** command) the SAG that you want to edit.

4. **sag-recursion**—Set this parameter to **enabled** if you want to use SIP SAG recursion for this SAG. The default is **disabled**.
5. **stop-sag-recurse**—Enter the list of SIP response codes that terminate recursion within the SAG. On encountering the specified response code(s), the Net-Net SBC returns a final response to the UAC. You can enter the response codes as a comma-separated list or as response code ranges; for example:

```
ACMEPACKET(session-agent-group)# stop-sag-recurse 404,407,484-486
```
6. **strategy**—Enter the strategy to determine how session agents will be chosen by this session-agent-group element. The following are valid values: hunt, roundrobin, leastbusy, propdist, lowsusrate.
7. Save and activate your configuration.

Configurable Route Recursion

When the Net-Net 9200 routes SIP requests from a UAC to a UAS, it might determine that there are multiple routes to try based on a matching local policy. The Net-Net 9200 recurses through the list of routes in a specific order according to your configuration and the quality of the match. There are other scenarios when a UAS replies with a 3xx Redirect response to the Net-Net 9200, the 3xx response can include multiple Contacts to which the request should be forwarded in a specific order. In both cases, the Net-Net 9200 needs to recurse through a list of targets.

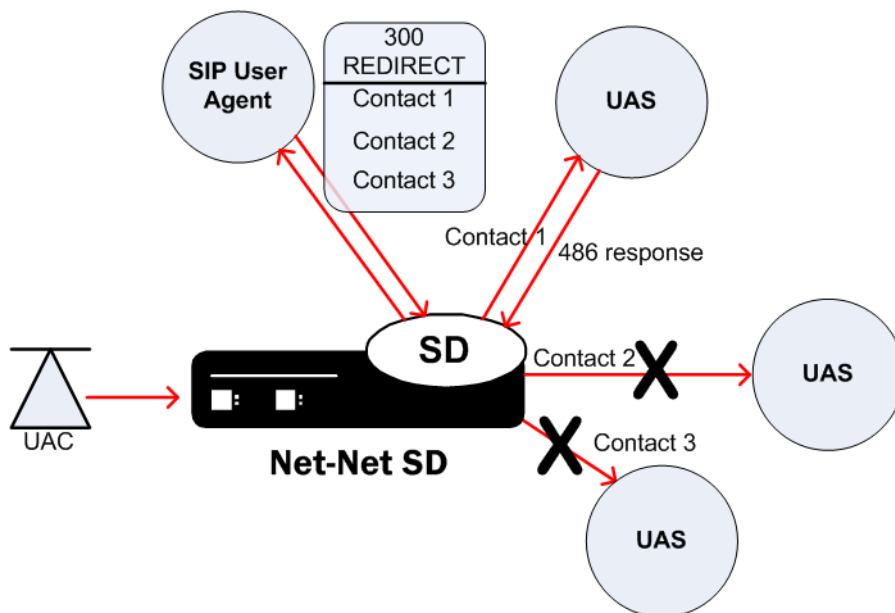
When the Net-Net 9200 receives a non-successful (or non-6xx response) final response from the UAS, and there are multiple targets for the original request, the 9200 will forward the request to the next target and wait for a response. While the process of forwarding the request to multiple targets as explained in the previous paragraph is called serial forking, and the process of forwarding the request to contacts received in redirect responses is called recursion, the term recursion is used for both processes in this notice.

Use the SIP Route Recursion feature when you want the Net-Net 9200 to forward a response to the UAC and stop recursing through the target list immediately after receiving the 3xx, 4xx, or 5xx response code that you configure. When this feature is disabled, the Net-Net 9200 only stops recursing when it receives a message with a 401 or 407 response code. Using this feature, you can configure a specific message or range of messages to stop recursing on when received. The Net-Net 9200 retains its default behavior to stop recursing on a 401 or 407 response code when SIP Route Recursion is configured on a SIP interface. The Net-Net 9200 will always stop recursing when it receives a global failure (6xx); this behavior is not configurable.

You can disable response recursion for either a SIP interface or for a SIP session agent, providing you with flexibility for various network architectures. For instance, a PSTN gateway might be the only hop to reach a given endpoint, whereas several session agents might need to be contacted if multiple devices map to a contacted address of record.

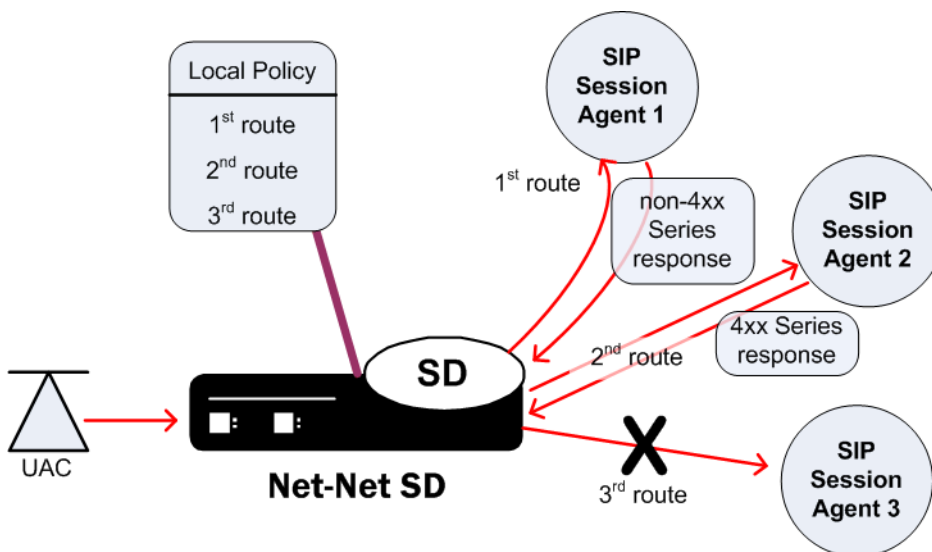
Example 1

A more detailed example is when a softswitch might return a list of contacts for multiple PSTN gateways in a Redirect message. If the PSTN target number contacted on redirection is busy, a 486 response will be sent to the Net-Net 9200. Since the single target is located in the PSTN, a subsequent request through a different gateway will yield another 486 response. The Net-Net 9200 should be configured to return the 486 response to the UAC immediately. No other SIP requests should be sent to applicable targets/contacts that were enumerated in the redirect list. See the following example:



Example 2

The Net-Net 9200 might determine from a local policy lookup that several routes are applicable for forwarding a SIP message. The Net-Net 9200 will try each route in turn, but the SIP response recursion disable feature can be implemented to stop the route recursion when a configured responses message is received by the Net-Net 9200. See the following example:



There are a few conditions on the parameter used to configure response recursion:

- SIP Route Recursion is configurable for either the SIP interface or session agent.
- 401 and 407 are preconfigured for all configured SIP interfaces. They are not configured for session agents.
- The format is a comma-separated list of response codes or response code ranges: 404, 484-486.
- Only response codes that fall within the 3xx, 4xx, and 5xx range may be specified.

ACLI Instructions and Examples

You enable SIP Route Recursion either in the session agent or the SIP interface configuration.

Configuring a Session Agent for SIP Route Recursion

To configure SIP Route Recursion for an existing session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router path.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. Select the session agent where you want this feature.


```
ACMEPACKET(session-agent)# select
<hostname>:
1: asd          realm=      ip=1.0.0.0
2: SIPSA        realm=      ip=10.10.102.1

selection:2
ACMEPACKET(session-agent)#
```
5. **stop-recurse**—Enter a comma-separated list of response codes or response code ranges, then and press <Enter>.


```
ACMEPACKET(session-agent)# stop-recurse 404,484-486
```
6. Save and activate your changes.

Configuring a SIP Interface for SIP Route Recursion

To configure SIP Route Recursion for an existing SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Select the SIP interface to which you want to apply this feature.


```
ACMEPACKET(sip-interface)# select
<realm-id>:
```

```
1: Acme_Realm
```

```
selection:1
```

```
ACMEPACKET(sip-interface)#
```

5. **stop-recurse**—Enter a comma-separated list of response codes or response code ranges, and then press <Enter>.

```
ACMEPACKET(sip-interface)# stop-recurse 404,484-486
```

6. Save and activate your changes.

Event Package Interoperability

Service providers often deploy a Net-Net 9200 on the border of an access network, where it sits between the SIP endpoints (user agents) and the service provider's application server. The application server and the user agents sometimes use various SIP event packages to exchange and maintain state information. The SUBSCRIBE and NOTIFY methods are used to establish subscriptions to the event packages and to report state changes to the subscribing entity.

The SIP global contact option addresses interoperability in the Dialog and Presence event packages that are used in hosted PBX and IP centrex offerings. State information is passed in the message body of a NOTIFY request; this message body is encoded in an XML format described by the Content-Type header. The Net-Net 9200 needs to update certain fields in the body to account for dialog mapping and SIP NAT functionality between the access and service provider realms. Often the subscriptions are established using URIs learned from Contact headers in the user agent registrations or dialog establishment (INVITE/SUBSCRIBE). For this, a Net-Net 9200 requires a Contact URI that is usable and routable outside of an existing dialog.

The SIP global contact option enables persistent URIs in the Contact headers inserted into outgoing SIP messages. If this option is not used, URIs placed in the Contact header of outgoing messages are only valid within the context of the dialog to which the message is associated.

RFCs associated with this feature are:

- A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June 2002
- J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, August 2004
- J. Rosenberg, et al. "Data Format for Presence Using XML," <http://www.iptel.org/info/players/ietf/presence/outdated/draft-rosenberg-impp-pidf-00.txt>, Work In Progress (expired), June 2000
- J. Rosenberg, H. Schulzrinne, R. Mahy, "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)," draft-ietf-sipping-dialog-package-06.txt, Work In Progress, April 2005
- H. Sugano, et al., "Presence Information Data Format (PIDF)," RFC 3863, August 2004

ACLI Instructions and Examples

This feature is applicable to the global SIP configuration.

To configure SIP event package interoperability:

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

- ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
 3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
 4. **options**—Add SIP event package interoperability support to a new SIP configuration or to an existing SIP configuration:
 - 4a. If you do not currently have an SIP configuration, you can add the option by typing **options**, a <Space> and then **global-contact**.


```
ACMEPACKET(sip-config)# options global-contact
```
 - 4b. Select the SIP configuration so that you can add SIP event package interoperability support to it. Then, to add this option to a list of options that you have already configured, type **options** followed by a <Space>, the plus sign (+), and the **global-contact** option.


```
ACMEPACKET(sip-config)# select
ACMEPACKET(sip-config)# options +global-contact
```

If you type **options global-contact** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.
 5. Save and activate your changes.

SIP REGISTER Forwarding After Call-ID Change

This feature addresses the case when an endpoint reboots and performs a third party registration before its old registration expires. During this reregistration, the contact header is the same as it was pre-reregistration. As a consequence of the reboot, the SIP Call-ID changes. In this situation, the Net-Net 9200 does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI.

To remedy this problem, the Net-Net 9200 now keeps track of the Call-ID in its registration cache. A new option in the sip-interface configuration element forces the Net-Net 9200 to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.

Instructions and Examples

To configure SIP REGISTER forwarding after a Call-ID change:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **options**—Add this feature to a new or an existing SIP configuration:

- 4a. If you do not currently have a SIP configuration, you can add the option by typing **options**, a <Space>, and then **forward-reg-callid-change**.
`ACMEPACKET(sip-config)# options forward-reg-callid-change`
- 4b. For an existing SIP configuration, select the SIP configuration so that you can add this feature to it. Then, to add this option to a list of options that you have already configured, type **options**, a <Space>, the plus sign (+), and the **forward-reg-callid-change** option.

`ACMEPACKET(sip-config)# options +forward-reg-callid-change`

If you type **options forward-reg-callid-change** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

5. Save and activate your changes.

SIP Local Response Code Mapping

The SIP local response code mapping feature (SIP response code translation profile in Net-Net EMS) has been added as an enhancement to the existing SIP response code mapping. The SIP response code map feature lets you establish a table that maps SIP response-received messages (entries) to response-to-send messages (entries).

SIP local response code mapping is used with the SIP responses generated by the Net-Net 9200 towards a specific SIP session agent. This feature lets you provision the mapping of the response codes used by the Net-Net 9200 when it generates the responses towards a session agent.

You create the SIP local response code map using the existing mapping functionality and then assigning that map to a session agent.

Note: The configured response map is not used when the Net-Net 9200 is acting as proxy for the responses to this session agent.

Instructions and Examples

The following instructions explain how to create the SIP response code map and then how to assign it to a specific session agent.

Creating a SIP Response Code Map

To create a SIP local response code map:

1. In Superuser mode, type **configure terminal** and press <Enter>.
`ACMEPACKET# configure terminal`
2. Type **session-router** and press <Enter>.
`ACMEPACKET(configure)# session-router`
`ACMEPACKET(session-router)#`
3. Type **sip-response-map** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
`ACMEPACKET(session-router)# sip-response-map`
`ACMEPACKET(response-map)#`
4. **name**—Enter the name of the SIP response map you want to configure. This value is required and must be unique.
`ACMEPACKET(response-map)# name busy`

5. **entries**—To configure the entries for this mapping, type **entries** and then press <Enter>. Typing a question mark will show you the response code entry parameters that you can configure.

```
ACMEPACKET(response-map)# entries
ACMEPACKET(response-map-entries)# ?
-----  ACLI v1.0  -----

recv-code  original SIP response code
xmit-code   translated SIP response code
reason      translated reason string
rhdr-protocol Reason header protocol type
rhdr-cause   Reason header cause value
rhdr-text    Reason header text
```

- 5a. **recv-code**—Enter original SIP response code for the **recv-code** parameter. This value is an integer from 100 to 866. Typing a question mark <?> will show you the short names and code values for all internally-generated 8xx series responses.
- 5b. **xmit-code**—Enter the SIP response code into which you want the original response code to be translated. This value is an integer from 100 to 699.
- 5c. **reason**—Enter a reason for the translated code into the reason parameter. This response comment is sent with the translated code. Make your entry in quotation marks.
- 5d. **rhdr-protocol**—Enter the protocol type to appear in the SIP reason header. Valid values are: SIP, q850, preemption.
- 5e. **rhdr-cause**—Enter the protocol cause value to appear after the “cause=” label of the SIP reason header.
- 5f. **rhdr-text**—Enter the text to appear after the “text=” label in the SIP reason header.
6. Note the name that you gave the SIP response code map so that you can use it when you configure a session agent to support SIP response code mapping.
7. Save and activate your changes.

Assigning SIP Response Code Maps to Session Agents

To use a SIP local response code map for a specific session agent configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **local-response-map**—Enter the name of configured SIP response map and press <Enter>.


```
ACMEPACKET(session-agent)# local-response-map busy
```
5. Save and activate your configuration.

Session Agent Ping Message Formatting

You can configure the user portion of the From: header, the To: header, and the Request-URI in the ping-type message that the Net-Net 9200 sends to a session agent. This feature is required for interoperability with certain E911 servers.

In the following example of a session agent ping-type message, you can set the user portion of the Request-URI (the text bob in the OPTIONS method line) and the user portion of the From: header (the text bob in the From: header) to the same new value. You can also set the user portion of the To: header (the text anna in the To: header) to its own new value.

```
OPTIONS sip:bob@sip.com SIP/2.0
From: UA1 <sip:bob@sip.com>
To: NUT <sip:anna@gw.sip.com>
Call-ID: 123abc@desk.sip.com
CSeq: 1 OPTIONS
Contact: <sip:UA1@client.sip.com>
Accept: application/sdp
Content-Length: 0
```

If you do not enable this feature, the session agent ping-type message will contain the text “ping” in all cases.

Instructions and Examples

To configure session agent ping message formatting:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-router configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding this feature to a pre-existing session agent configuration, then you need to select the configuration before editing it.
4. **ping-from-user-part**—Enter the *user and user URI portion* for the From user part.


```
ACMEPACKET(session-agent)# ping-from-user-part bob
```
5. **ping-to-user-part**—Enter the *user* for the From user part.


```
ACMEPACKET(session-agent)# ping-to-user-part anna
```
6. Save and activate your configuration.

SIP PPI—PAI Conversion

The SIP PAI stripping offers the ability to covert P-Preferred-Identity and P-Asserted-Identity (PAI) headers so that service providers can ensure an extra measure of security against malicious users pretending to be legitimate users.

Instructions and Examples

There are four valid values for the **ppi-to-pai** parameter:

- **none**—The Net-Net 9200 performs no conversion
- **trusted**—The Net-Net 9200 checks the configured trust level for the ingress endpoint; if it is untrusted, then PPI headers in the message are deleted; if it is trusted, then PPIs are converted to PAIs
- **remove**—The Net-Net 9200 removes PPI headers
- **always**—The Net-Net 9200 always converts PPIs to PAIs

To configure SIP PAI stripping for an existing realm using the ACLI:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **ppi-to-pai**—Enter the conversion mode that you want to use; valid values appear in the description above.

Configuring SIP Statuses to Q.850 Reasons

This section explains the Net-Net 9200's ability to map Q.850 cause values with SIP responses, a feature used in SIP calls.

RFC 3326 defines a header that might be included in any in-dialogue request. This reason header includes cause values that are defined as either a SIP response code or ITU-T Q.850 cause values. You can configure the Net-Net 9200 to support sending and receiving RFC 3326 in SIP messages for:

- Mapping SIP response messages and RFC 3326 reason header and cause
- Locally generated SIP response with RFC 3326 reason header and cause

As specified in RFC 3326, the Net-Net 9200 sends SIP responses to the softswitch that contain the received Q.850 cause code and the reason.

Though the Net-Net 9200 can generate RFC 3326 headers, the default behavior for this feature is disabled. Furthermore, the Net-Net 9200 can receive and pass SIP error messages (4xx, 5xx, and 6xx) that contain the SIP reason header with a Q.850 cause code and reason (as specified in RFC 3326). If the Net-Net 9200 receives an error message without the Reason header, then the Net-Net 9200 is not required to insert one.

SIP-SIP Calls

The SIP Reason header might appear in any request within a dialog, in a CANCEL request, and in any response where the status code explicitly allows the presence of this header field. The syntax of the header follows the standard SIP parameter:

```
Reason: SIP;cause=200;text="completed elsewhere"
Reason: Q.850;cause=16;text="Terminated"
```

This feature attends to the following possible SIP call scenarios:

- When the Net-Net 9200 receives a SIP request or SIP response that contains the Reason header, the Net-Net 9200 passes it without modification.
- When it generates a SIP response, the Net-Net 9200 includes the RFC 3326 Reason header containing a Q.850 cause code and reason. This is the case for all local conditions and for all internally generated error responses (4xx, 5xx, and 6xx) to an initial SIP INVITE.

Possible local error scenarios are:

- invalid-message
- cpu-overloaded
- media-released
- media-not-allocated

ACLI Instructions and Examples

Configuration for reason cause mapping for SIP-SIP calls requires that you set up the ACLI local-response-map configuration with appropriate entries; these generate the SIP response and the Q.850 cause code value to be used for particular error scenarios.

If you want to add a Reason header, then you need to enable that capability in the global SIP configuration.

To configure a local response map:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **local-response-map** and press <Enter>.

```
ACMEPACKET(session-router)# local-response-map
ACMEPACKET(local-response-map)#
```

4. Type entries and press <Enter>.

```
ACMEPACKET(local-response-map)# entries
ACMEPACKET(local-response-map-entry)#
```

From here, you can view the entire menu for the local response map entries configuration by typing a ?.

5. **local-error**—Set the local error that triggers the use of this local response map. Your choices are: *invalid-message*, *cpu-overload*, *media-released*, and *media-not-allocated*.
6. **sip-status**—Set the SIP response code to use. There is no default, and the valid range for values is 100-699.
7. **sip-reason**—Set the SIP reason string you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.

8. **q850-cause**—Set the Q.850 cause. There is no default value.
9. **q850-reason**—Set the Q.850 reason string that you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
10. Repeat this process to create the number of local response map entries that you need.
11. Save and activate your configuration for changes to take effect.

To enable the Net-Net 9200 to add the Reason header:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **add-reason-header**—Set this parameter to **enabled** to turn add the Reason header. The default is **disabled**.
5. Save and activate your configuration for changes to take effect.

Default Mappings

This table defines the default mappings the Net-Net 9200 uses when it cannot locate an appropriate entry that you have configured.

Q.850 Cause Value		SIP Status		Comments
1	Unallocated number	404	Not found	
2	No route to specified transit network	404	Not found	
3	No route destination	404	Not found	
16	Normal calling clearing		BYE message	A call clearing BYE message containing cause value 16 normally results in the sending of a SIP BYE or CANCEL request. However, if a SIP response is to be sent to the INVITE request, the default response code should be used.
17	User busy	486	Busy here	
18	No user responding	408	Request timeout	
19	No answer from the user	480	Temporarily unavailable	
20	Subscriber absent	480	Temporarily unavailable	

Q.850 Cause Value		SIP Status		Comments
21	Call rejected	603	Decline (if location filed in Cause information element indicates user; otherwise 403 Forbidden is used)	
22	Number changed	301	Moved permanently (if information in diagnostic field of Cause information element is suitable for generating SIP Contact header; otherwise 410 Gone is used)	
23	Redirection to new destination	410	Gone	
25	Exchange routing error	483	Too many hops	
27	Destination out of order	502	Bad gateway	
28	Address incomplete	484	Address incomplete	
29	Facility rejected	501	Not implemented	
31	Normal, unspecified	480	Temporarily unavailable	
34	No circuit, channel unavailable	503	Service unavailable	
38	Network out of order	503	Service unavailable	
41	Temporary failure	503	Service unavailable	
42	Switching equipment congestion	503	Service unavailable	
47	Resource unavailable unspecified	503	Service unavailable	
55	Incoming calls barred with CUG	403	Forbidden	
57	Bearer capability not authorized	403	Forbidden	
58	Bearer capability not presently available	503	Service unavailable	
65	Bearer capability not implemented	488	Not acceptable here	
69	Requested facility not implemented	501	Not implemented	

Q.850 Cause Value		SIP Status		Comments
70	Only restricted digital information available	488	Not acceptable here	
79	Service or option not implemented, unspecified	501	Not implemented	
87	User not member of CUG	403	Forbidden	
88	Incompatible destination	503	Service unavailable	
102	Recovery on timer expiry	504	Server time-out	

ACLI Instructions and Examples

To configure a SIP status to Q.850 Reason with cause mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-q850-map** and press <Enter>.


```
ACMEPACKET(session-router)# sip-q850-map
ACMEPACKET(sip-q850-map)#
```
4. Type **entries** and press <Enter>.


```
ACMEPACKET(sip-q850-map)# entries
ACMEPACKET(sip-q850-map-entry)#
```

From here, you can view the entire menu for the SIP status to Q.850 Reason with cause mapping entries configuration by typing a ?.
5. **sip-status**—Set the SIP response code that you want to map to a particular Q.850 cause code and reason. There is no default, and the valid range for values is 100-699.
6. **q850-cause**—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
7. **q850-reason**—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

To configure a Q.850 cause to a SIP status with reason mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-q850-map** and press <Enter>.

```
ACMEPACKET(session-router)# q850-sip-map
```

```
ACMEPACKET(q850-sip-map)#
```

4. Type **entries** and press <Enter>.

```
ACMEPACKET(q850-sip-map)# entries
```

```
ACMEPACKET(q850-sip-map-entry)#
```

From here, you can view the entire menu for the Q.850 cause to a SIP response code with reason mapping entries configuration by typing a ?.

5. **q850-cause**—Set the Q.850 cause code that you want to map to a SIP status with reason. There is no default.
6. **sip-status**—Set the SIP response code to which you want to map the Q.850 cause that you set in step 5. There is no default, and the valid range for a value is 100-699.
7. **sip-reason**—Set the reason that you want to use with the SIP response code that you specified in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

Trunk Group URIs

The Net-Net 9200's trunk group URI feature, applicable for SIP signaling services, enables the capabilities related to trunk groups that are described in this section. This implementation follows the IPTTEL draft "Representing Trunk Groups in Tel/SIP Uniform Resource Identifiers (URIs)" (draft-ietf-iptel-trunk-group-06.txt), and also supports more customized approaches.

- For a typical access call flow scenario, when the calling party's call arrives at the Net-Net 9200, the Net-Net 9200 formulates a SIP INVITE message that it sends to a softswitch. The Net-Net 9200 now supports a new URI contact parameter in the SIP request message so that service providers need to be able to:
 - Determine from where the Net-Net 9200 received the call
 - Signal information about the originating gateway from a Net-Net 9200 to a softswitch (e.g., an incoming trunk group or a SIP gateway to a Net-Net 9200)
- This feature supports the signaling of routing information to the Net-Net 9200 from network routing elements like softswitches. This information tells the Net-Net 9200 what egress route (or outgoing trunk groups) it should choose for terminating next hops/gateways. For this purpose, new SIP URI parameters in the Request-URI are defined. Additional URI parameters include the network context to identify the network in which the originating or terminating gateway resides.
- Especially important for large business applications, this feature can free Net-Net 9200 resources by reducing the number of local policy, session agent, and session agent group configurations. By enabling the trunk group URI feature, the Net-Net instead uses a routing scheme based on signaled SIP URI information.

Terminology

The following IPTTEL terms are used in the descriptions of and instructions for how to configure this feature:

- **Trunk**—In a network, a communication path connecting two switching systems used in the establishment of an end-to-end connection; in selected applications, it may have both its terminations in the same switching system
- **Trunk group**—A set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable except where sub-grouped
- **Trunk group name**—Provides a unique identifier of the trunk group; referred to as tgrp
- **Trunk group context**—Imposes a name space by specifying a domain where the trunk groups are; also referred to simply as “context”

Trunk Group URI Parameters

Trunk group URI parameters identify originating and terminating trunk group information in SIP requests.

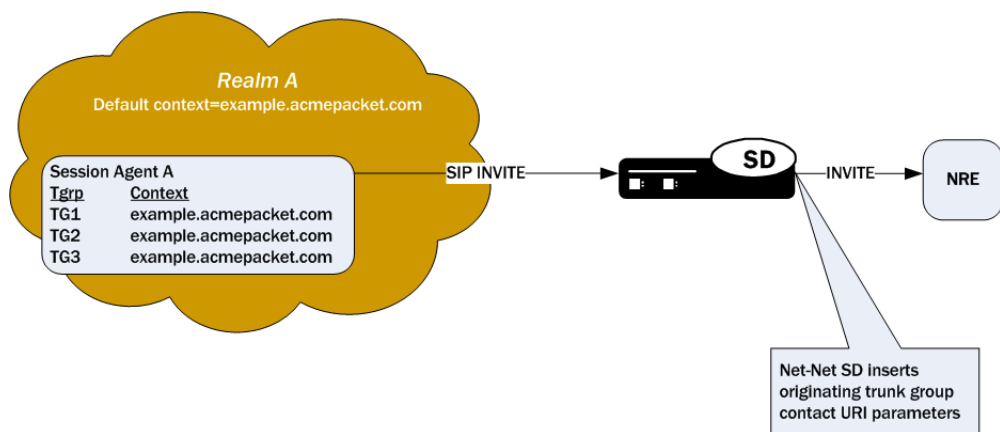
In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net 9200 allows you to define URI parameters for use with originating and terminating trunk group URIs.

Originating Trunk Group URI Parameters and Formats

You can configure session agents and session agent groups on the Net-Net 9200 to insert trunk group URI parameters in the SIP contact header. When SIP gateways comply with the IPTEL draft, they include the originating URI parameter in the SIP contact header. For those SIP gateways that are not compliant, the Net-Net 9200 inserts SIP trunk group URI parameters on the gateway's behalf.

When there are no applicable session agent or session agent group configurations, the Net-Net 9200 uses the source IP address of the endpoint or gateway as the trunk group name (tgrp) parameter in the originating trunk group URI.

The following diagram shows a scenario where the Net-Net inserts originating trunk group URI parameters.



There are two available formats for the originating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (identifier of the specific trunk group) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:
 - tgrp="trunk group name"

- trunk-context="network domain"
- The URI BNF for would appear as it does in the example directly below, where the tgrp is tg55 and the trunk-context is trunk-context is telco.example.com:

tel:+15555551212;tgrp=tg55;trunk-context=telco.example.com

2. The second format is customized specifically for access URIs and contains two provisioned parameters: tgrp (or tgroupName) and context (or provstring). This appears as tgrp.context, where these definitions apply:

- tgrp (tgroupName)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
- context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

custsite2NY-00020.type2.voip.carrier.net

where the tgrp is custsite2NY-00020, and the context is type2.voip.carrier.net.

The BNF for an access URI conforms to the following:

```
SIP-URI = "sip:" [userinfo] hostport uri-parameters [headers]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param
```

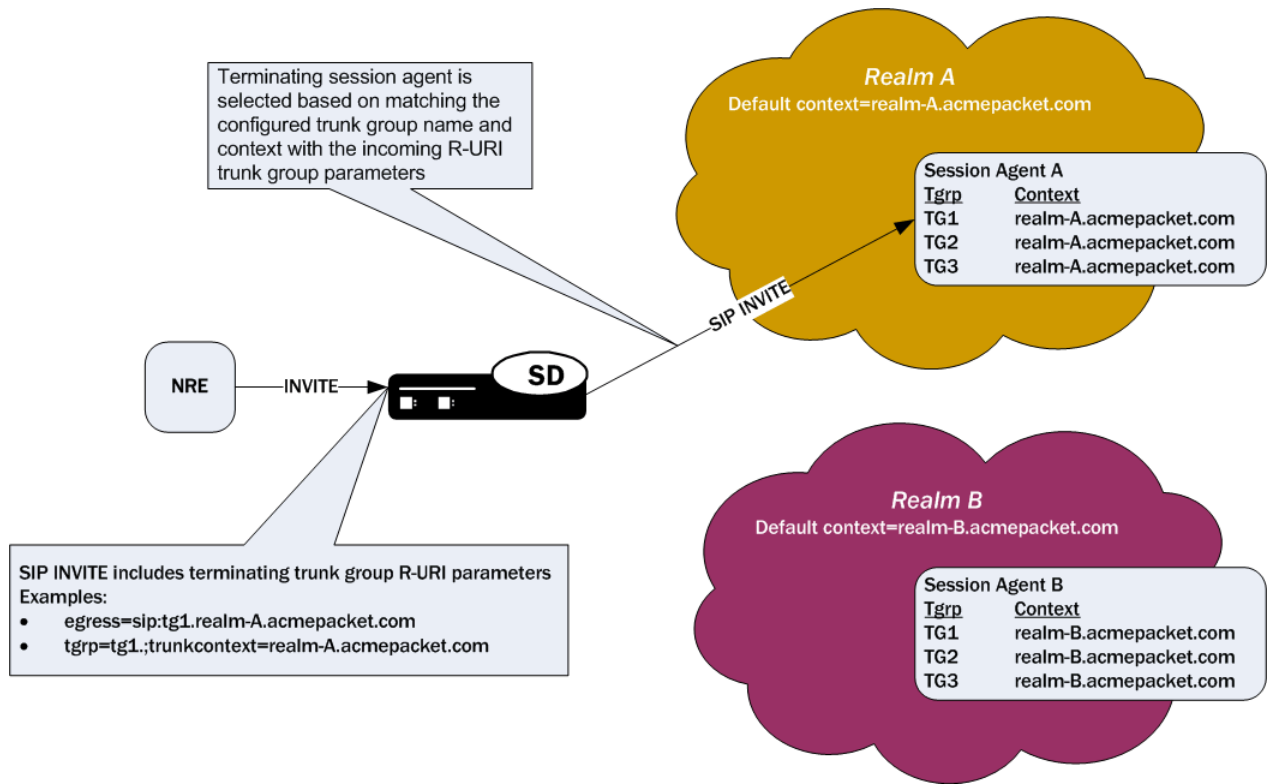
```
other-param = accessid / pname [ '=' pvalue ]
accessid = "access=" accessURI
```

```
accessURI = scheme tgroupName [ "." provstring ]
scheme = "sip:" / token
tgroupName = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
provstring = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-") alphanum
domain = alphanum / alphanum *( alphanum / "-") alphanum
```

Terminating Trunk Group URI Parameters and Formats

Terminating trunk group URI parameters appear in the R-URI, and they can be included in by a network routing element to instruct the Net-Net 9200 which egress trunk groups to use. By matching the trunk group URI parameter with configured session agents or session agent groups, the Net-Net 9200 can locate the terminating

gateway. The trunk group name can also be expressed as the IP address of the terminating gateway.



In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net allows you to define the URI parameters used in terminating trunk groups.

There are two available formats for the terminating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (which can be either a trunk group name or an IP address) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

- tgrp="trunk group name"
- trunk-context="network domain"

An example R-URI with terminating trunk group parameters appears as follows, where the tgrp is TG2-1 and the context is isp.example.net@egwy.isp.example.net:

```
INVITE sip:+15555551212;tgrp=TG2-1;trunk-context=isp.example.net@egwy.isp.example.net SIP/2.0
```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: tgrp (or tgrname) and context (or tgdomain). This appears as tgrp.context (or tgrname.tgdomain), where definitions apply:
 - tgrp (tgrname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot

contain a period (.), and can contain a hyphen (-) but not as the first or the last character

- context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```

SIP-URI = "sip:" [userinfo ] hostport uri-parameters [headers ]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param

other-param = egressid / pname [ '=' pvalue ]
egressid = "egress=" egressURI
egressURI = scheme tname [ "." tgdomain ]
scheme = "sip:" / token
tname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
tgdomain = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-") alphanum
domain = alphanum/ alphanum *( alphanum / "-") alphanum

```

How It Works

For all trunk group URI support, you must set the appropriate parameters in the SIP manipulations configuration and in the session agent or session agent group configurations.

In the originating trunk group URI scenario, a call arrives at the Net-Net 9200 from a configured session agent or session agent group. If this session agent or session agent group has the appropriate trunk group URI parameters and inbound manipulation rules configured, the Net-Net 9200 then looks to the SIP manipulations configuration and add the trunk group URI information according to those rules. Those rules tell the Net-Net 9200 where and how to insert the trunk group URI information, and the Net-Net 9200 forwards the call.

In the terminating trunk group scenario, a call arrives at the Net-Net 9200 from, for instance, a call agent. This call contains information about what trunk group to use. If the information matches a session agent or session agent group that has outbound manipulation rules configured, the Net-Net 9200 will then look up the SIP manipulations configuration and strip information according to those rules. Those rules tell the Net-Net 9200 where and how to remove the information, and the Net-Net 9200 forwards the call.

SIP Header and Parameter Manipulation

SIP header and parameter manipulation is its own configuration where you can set up rules for the addition, removal, and modification of a SIP header or the elements of a SIP header. For example, you can set up the configuration to add a URI parameter to the URI in a SIP header or replace an FQDN with an IP address. For trunk group URI support, this configuration tells the Net-Net 9200 where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

These manipulations can be applied at the realm or at the session agent level.

To learn more about SIP header manipulation, refer to the “SIP Header and Parameter Manipulation” section of this guide’s *SIP Services* chapter.

Trunk Group Routing

You can configure SIP interfaces (using the ACLI **term-tgrp-mode** parameter) to perform routing based on the trunk group information received in SIP requests. There are three options: none, IPTEL, and egress URI.

- If you leave this parameter set to none (its default), the Net-Net 9200 will not look for or route based on terminating trunk group URI parameters
- When you set this parameter to either **iptel** or **egress-uri** and the incoming request has the trunk group parameter of this type (IPTEL or egress URI), the Net-Net 9200 will select the egress next hop by matching the “tgrp” and “trunk context” with a configured session agent or session agent group.

If the received terminating trunk group URI parameters include an IP address, the egress next hop is the IP address specified. The Net-Net 9200 determines the egress realm by matching the trunk context it receives with the trunk context you configure for the realm.

- If the incoming request does not have trunk group parameters or it does not have trunk group parameters of the type that you configure, the Net-Net 9200 uses provisioned procedures and/or local policy for egress call routing.

The Net-Net 9200 returns errors in these cases:

- If the terminating trunk group URI parameters do not identify a local Net-Net 9200 session agent or session agent group, then the Net-Net 9200 returns a SIP final response of “488 Not Acceptable Here.”
- If the Net-Net 9200 receives a SIP INVITE with terminating trunk group URI parameters that do not match the specified syntax, the Net-Net 9200 returns a 400 final response with the reason phrase Bad Egress = Parameters.

Trunk Group URIs and SIP Registration Caching

For calls where SIP registration caching is used, you will need to set certain parameters that enable the Net-Net 9200 to preserve trunk group URI parameters on the outgoing side.

- For SIP-SIP calls, you set the **preserve-user-info** option in the SIP interface configuration.

ACLI Instructions and Examples

Before you configure your Net-Net 9200 to support trunk group URIs, you need to determine:

- How you want to manipulate SIP headers (entered in the SIP header manipulations configuration)
- For terminating trunk group routing, the trunk group mode you want to use (none, IPTEL, or egress URI); this decides routing based on trunk group information
- The trunk group name and context to use entered in a session agent or session agent group configuration
- Whether you are using originating or terminating trunk group URIs (entered in the session agent configuration)
- The trunk group context for use in a realm configuration, in case the trunk group name in the session agent or session agent group does not have a context

Configuring SIP Manipulations

For detailed instructions about how to configure SIP header a manipulations, refer to the “SIP Header and Parameter Manipulation” section of this guide’s *SIP Services* chapter.

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The **new-value** parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are \$TRUNK_GROUP and \$TRUNK_GROUP_CONTEXT

Setting the Trunk Group URI Mode for Routing

To set the mode for routing for terminating trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **term-tgrp-mode**—Set the mode that you want to use. Your choices are:
 - none—Disables routing based on trunk groups
 - iptel—Uses trunk group URI routing based on the IPTEL formats
 - egress-uri—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **out-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the Net-Net 9200 via this session agent. There is no default.
5. **in-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic entering the Net-Net 9200 via this session agent. There is no default.
6. **trunk-group**—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context,

then the Net-Net 9200 will use the one you set in the realm for this session agent.

Your ACLI entries for this list must one of these formats: `tgrp:context` or `tgrp.context`.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent)# trunk-group (tgrp1:context1 tgrp2:context2)
```

Configuring a Session Agent Group for Trunk Group URIs

In a session agent group, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent group for trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-group** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent-group
ACMEPACKET(session-agent-group)#
```
4. **trunk-group**—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Net-Net 9200 will use the one you set in the realm for this session agent group.

Your ACLI entries for this list must take one of these formats: `tgrp:context` or `tgrp.context`.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent-group)# trunk-group (tgrp1:context1 tgrp2:context2)
```

Setting a Trunk Group Context in a Realm

You can set trunk group contexts at the realm level, which will be used by all session agents and session agent groups if there is no context specified in their configurations.

The realm trunk group URI context accommodates the IPTEL and the custom format.

To configure a trunk group context for a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **trunk-context**—Enter the trunk group context to use for this realm. There is no default.

Using this Feature with a SIP Interface

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the **preserve-user-info** option for that SIP interface.

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-group** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **options**—Add support for trunk group URIs with SIP interface that uses registration caching.

If you are adding this option to a new SIP interface, you can just type **options**, a <Space>, and **preserve-user-info**.

If are adding this to an existing SIP interface, you must type a “plus” (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +preserve-user-info**.

**Example 1: Adding
Originating Trunk
Group Parameters in
IPTTEL Format**

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTTEL format.

```

sip-manipulation
  name
  header-rule
    name
    action
    match-value
    msg-type
    element-rule
      name
      type
      action
      match-val-type
      match-value
      new-value
  element-rule
    name
    type
    action
    match-val-type
    match-value
    new-value
  add_iptel
    contact
    manipulate
    any
    tgrp
    uri-user-param
    add
    any
    $TRUNK_GROUP
    trunk-context
    uri-user-param
    add
    any
    $TRUNK_GROUP_CONTEXT

```

**Example 2: Adding
Originating Trunk
Group Parameters in
Custom Format**

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in custom format.

```

sip-manipulation
  name
  header-rule
    name
    action
    match-value
    msg-type
    element-rule
      name
      type
      action
      match-val-type
      match-value
      new-value
  add_att
    contact
    manipulate
    any
    egressURI
    uri-param
    add
    any
    "sip:"+$TRUNK_
    GROUP+"."+"$TRUNK
    _GROUP_CONTEXT

```

Example 3: Removing IPTEL Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove IPTEL trunk groups names.

```

sip-manipulation
  name
  header-rule
    name
    action
    match-value
    msg-type
    element-rule
      name
      type
      action
      match-val-type
      match-value
      new-value
    element-rule
      name
      type
      action
      match-val-type
      match-value
      new-value
  strip_ipTEL
    request-uri
    manipulate
    any
    tgrp
    uri-user-param
    delete-element
    any
    trunk-context
    uri-user-param
    delete-element
    any

```

Example 4: Removing Custom Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove custom trunk groups names.

```

sip-manipulation
  name
  header-rule
    name
    action
    match-value
    msg-type
    element-rule
      name
      type
      action
      match-val-type
      match-value
      new-value
  strip_egress
    request-uri
    manipulate
    any
    egressURI
    uri-param
    delete-element
    any

```

SIP Early Media Suppression

This section explains how to configure SIP early media suppression, which lets you determine who can send early media and in what direction. Early media are the RTP/RTCP packets sent from the called party to the caller, or vice versa, before a session is fully established (before a 200 OK is received). When the Net-Net SBC receives an INVITE message with SDP, it can forward media packets to the calling endpoint as soon as it forwards the INVITE to the next hop. It can also forward media packets received from the calling endpoint to the called endpoint as soon as the Net-Net SBC receives SDP in a SIP response to the INVITE, usually a provisional

message. This allows for any early media to be played, such as remote ringback or announcement.

Early media can be unidirectional or bidirectional, and can be generated by the caller, the callee, or both.

With early media suppression, you can block early media until the call is established. You can define which outbound realms or next hop session agents are allowed to send or receive early media. early media suppression only applies to RTP packets. RTCP packets received by Net-Net SBC are still forwarded to their destination in both directions, unless an endpoint is behind a NAT and the media manager has not been enabled for RTCP forwarding.

Note: To use early media suppression, you cannot configure media release of any kind: same-realm, same-network, or multiple-system media release.

How It Works

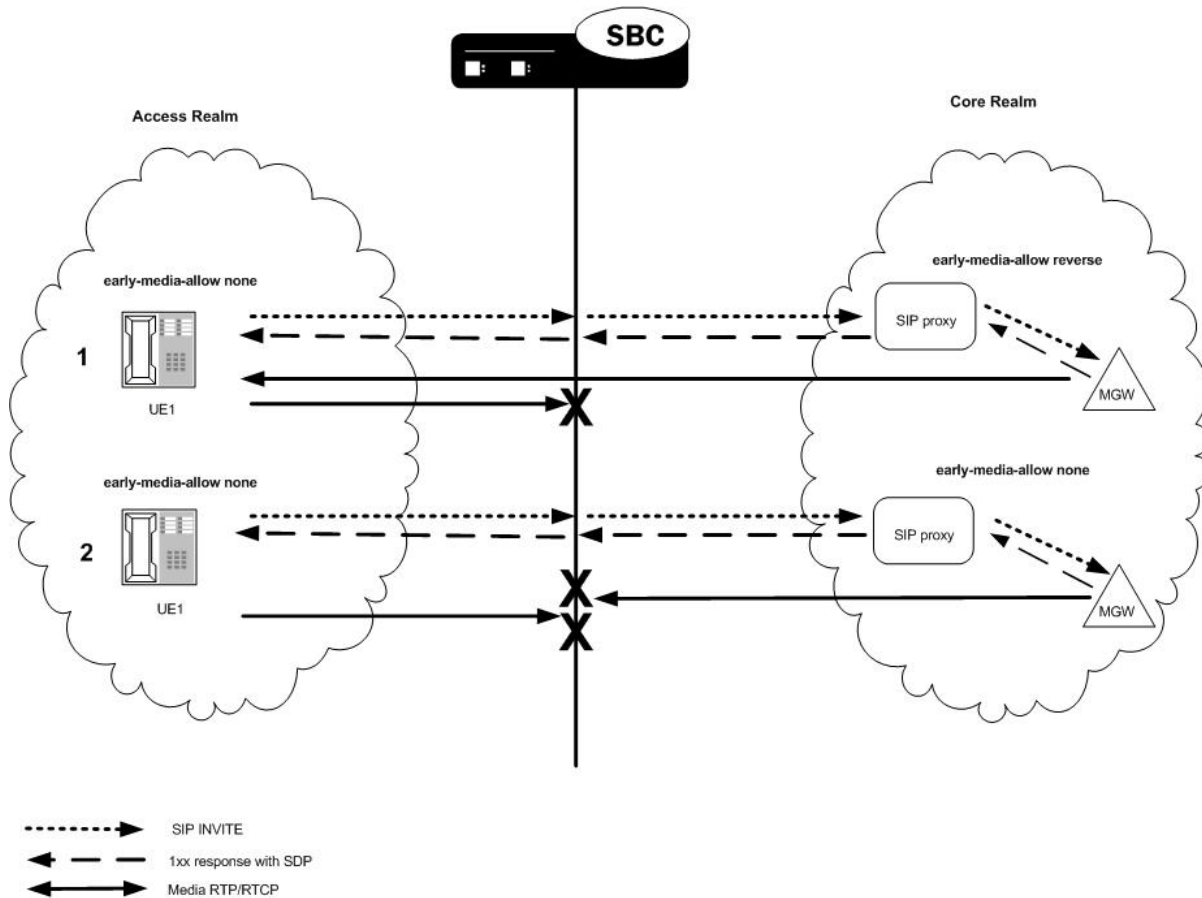
With the SIP-based addressing, early media suppression is based on the outbound SIP interface realms and the value of their early-media-allow parameter. When the Net-Net SBC forwards a SIP Invite out a SIP interface, the outbound realm is chosen based on the SIP layer information, such as the session agent for the next-hop or the address prefix of the next-hop SIP device. The matching realm's early-media-allow parameter value then applies to either allow all, block all, or block one-way early media until a 200 ok is received. At that point bidirectional media is allowed. The decision is based on SIP-layer addressing of next-hops.

You configure a rule for a realm or a session agent to use early media suppression. An early media suppression rule specifies whether you want to prevent early media in any direction, allow early media going to the calling endpoint in the reverse direction, or allow early media in both directions. The forward direction is when the packets flow from the caller to the called party. The reverse direction is when the packets flow from the called party to the caller.

The early media suppression rule is applied to a session. When the Net-Net SBC initiates a new session, it first checks whether the next hop is a session agent and if so, whether an early media suppression rule has been configured it. If an early media suppression rule is found, the Net-Net SBC enforces it. If the next hop is not a session agent or no early media suppression rule is configured, the Net-Net SBC checks whether an early media suppression rule has been configured for the outbound realm. If it finds one, it enforces it.

Example

The following illustration shows two examples of early media suppression.



1. Caller UE1 makes a call to the PSTN media gateway (MGW). The INVITE traverses from UE1 to the Net-Net SBC through the softswitch to the MGW. The Net-Net SBC allows early media from the core to reach UE1.
2. The PSTN MGW makes a call to UE1. The INVITE traverses to the Net-Net SBC and to UE1. The Net-Net SBC blocks all early media to and from UE1 until a 200 OK is received.

Early Media Suppression Support

The Net-Net SBC supports suppressing early media in the following directions no matter which side makes the SDP offer, until it receives 200 OK for an INVITE:

- Forward direction based on the outbound realm or next-hop session agent
- Forward and reverse directions based on the outbound realm or next-hop session agent.

The Net-Net SBC allows all media when a 200 OK response is received for the INVITE, regardless of whether the 200 OK response contains SDP.

Call Signaling

The Net-Net SBC media manager performs early media suppression according to an early media suppression rule. No change has been made to call signaling. For SIP, the Net-Net SBC still forwards SDP received in an INVITE request or response after performing a NAT to the media connection address. After which, the Net-Net SBC

is ready to receive media packets from the endpoints. If an early media suppression rule has been configured, the Net-Net SBC drops the packets going in the direction being specified by the rule.

For a H.323 to SIP call, early media suppression rule does not change how the Net-Net SBC performs H.225/Q.931 call signaling and starts the H.245 procedure (if required) to establish logical channels for early media on the H.323 leg of the call.

Suppression Duration

When early media suppression is enabled in a session, the block lasts until the session is established. For a SIP to SIP call or an H.323 to SIP call, a session is established when the Net-Net SBC receives a 200 OK response to the INVITE. A 200 OK response to the INVITE terminates early media suppression, even when it does not contain a SDP. (A 200 OK response to a PRACK or an UPDATE request does not terminate early media suppression.) After a session is established, the Net-Net SBC can receive a change in media session (for example, a re-INVITE with a new SDP) without an early media suppression rule blocking the media.

About the Early Media Suppression Rule

An early media suppression rule is configured in the form of a permission. It specifies whether early media is allowed in both directions, the reverse direction only or not at all. Reverse direction media is media sent in the upstream direction towards the calling endpoint.

Session Agent Rule

The next-hop session agent's early media suppression rule is applied regardless of whether the media packet's source or destination address is the same as the session agent's address. For example, if the session's next hop session agent is 10.10.10.5 but the SDP in a 183 response specifies 10.10.10.6 as its connection address.

Rule Resolution

When the call's next hop is a session agent and both the outbound realm of the call and the session agent have an early media suppression rule, the session agent's early media suppression rule takes precedence. If the session agent's early media suppression rule has not been configured, the outbound realm's early media suppression rule is used, if configured.

Selective Early Media Suppression

Normally, the Net-Net SBC performs early media blocking based on destination realm. Calls to such realms are prohibited from sending and receiving RTP until a SIP 200 OK response is received, and you can set the direction of the blocked media.

While decisions to block early media are customarily based on SIP-layer addressing, there are cases when the Net-Net SBC can reject early media based on the SDP address in the SDP answer for a 1XX or 2XX response. By comparing the SDP address with the realm prefix or additional prefix address, it can block early media for matching realms. For these cases, you define global or signaling realms—ones that are not tied to SIP interfaces, but which establish additional address prefixes and rules for blocking early media.

This way, the Net-Net SBC blocks all early media for SIP interface realms, but can accept it for global realms that reference media or PSTN gateways. This configuration allows early media for calls destined for the PSTN, and blocks it for user-to-user and PSTN-to-user calls.

Selective early media suppression addresses the fact that some service providers need to allow early media for certain user-to-user and PSTN-to-user calls to support, for example, custom ringback tones. The enhancements also address the fact that Net-Net SBCs can themselves lose the ability to decide whether or not early media should be blocked when confronted with hairpinned call flows, or with traffic that traverses multiple Net-Net SBCs.

How It Works

To address this need, you can configure realm groups. Realm groups are sets of source and destination realms that allow early media to flow in the direction you configure. For example, you can set up realm groups to allow media from PSTN realms to user realms so that users can listen to PSTN announcements, but prohibit early media from user realms to PSTN realms.

Note that the source and destination realms you add to your lists need to be a global or signaling realm matching the caller's SDP address prefix or a SIP realm.

Configuring Early Media Suppression

If the Net-Net SBC is serving as a media bridge, see *Configuring Media Suppression for Media Bridge* for information.

Configuring the Realm

To configure the realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. If configuring an existing realm, enter the select command to select the realm.
5. **early-media-allow**—Enter the early media suppression rule for the realm. The valid values are:
 - **none**—No early media is allowed in either direction
 - **both**—Early media is allowed in both directions
 - **reverse**—Early media received by Net-Net SBC in the reverse direction is allowed

There is no default value. If you leave this parameter blank, early media is allowed in either direction. You can use the following command to clear this parameter:

```
early-media-allow ()
```

6. Save and activate your configuration.

For example:

```
realm-config
    identifier          access1
    addr-prefix         192.168.1.0/24
```

```

network-interfaces
media:0
mm-in-realm          enabled
mm-in-network        enabled
msm-release          disabled
qos-enable           disabled
max-bandwidth        0
max-latency          0
max-jitter           0
max-packet-loss      0
observ-window-size   0
parent-realm
dns-realm
media-policy
in-translationid
out-translationid
class-profile
average-rate-limit   0
access-control-trust-level none
invalid-signal-threshold 0
maximum-signal-threshold 0
deny-period          30
early-media-allow     none
last-modified-date    2006-02-06 13:09:20

```

Configuring Session Agents

If you do not configure early media suppression for a session agent, the early media suppression for the outbound realm is used, if configured.

To configure session agents:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. If configuring an existing session agent, enter the select command to select the session agent.
5. **early-media-allow**—Enter the early media suppression rule for the session agent. The valid values are:
 - **none**—No early media is allowed in either direction
 - **both**—Early media is allowed in both directions
 - **reverse**—Early media received by Net-Net SBC in the reverse direction is allowed

There is no default value. If you leave this parameter blank, early media is allowed in either direction. You can use the following command to clear this parameter:

```
early-media-allow ()
```

6. Save and activate your configuration.

For example:

```
session-agent
  hostname                cust1
  ip-address              192.168.1.24
  port                    5060
  state                   enabled
  app-protocol            SIP
  app-type
  transport-method        UDP
  realm-id                access1
  description
  carriers
  allow-next-hop-lp       enabled
  constraints             disabled
  max-sessions            0
  max-outbound-sessions   0
  max-burst-rate          0
  max-sustain-rate        0
  time-to-resume          0
  ttr-no-response         0
  in-service-period       0
  burst-rate-window       0
  sustain-rate-window     0
  req-uri-carrier-mode    None
  proxy-mode
  redirect-action
  loose-routing           enabled
  response-map
  ping-method
  ping-interval           0
  media-profiles
  in-translationid
  out-translationid
  trust-me                disabled
  early-media-allow       reverse
  last-modified-date      2006-05-06 13:26:34
```

Configuring Realm Groups

To configure a realm group for selective early media suppression:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-group** and press <Enter>.


```
ACMEPACKET(media-manager)# realm-group
ACMEPACKET(realm-group)#
```
4. **name**—Enter the name of the realm group.

5. **source-realm**—Enter the list of one or more global/SIP realms that you want to designate as source realms for the purpose of blocking early media; this is the realm identifier value for the realms you want on the list. Values in this list refer to calling SDP realms; this parameter has no default. To enter more than one realm in the list, list all items separated by a comma and enclose the entire entry in quotation marks:

```
ACMEPACKET(realm-group)# source-realm "Private, Public"
```

To add a realm to the list, use the plus sign (+) in front of each new entry.

```
ACMEPACKET(realm-group)# source-realm +Private
```

You can also remove single items in the list by using the minus sign (-) directly in front of the realm identifier.

```
ACMEPACKET(realm-group)# source-realm -Private
```

6. **destination-realm**—Enter the list of one or more global/SIP realms that you want to designate as destination realms for the purpose of blocking early media; this is the realm identifier value for the realms you want on the list. Values in this list refer to called SDP realms; this parameter has no default. To enter more than one realm in the list, list all items separated by a comma and enclose the entire entry in quotation marks:

7. **ACMEPACKET(realm-group)# source-realm "Private, Public"**

To add a realm to the list, use the plus sign (+) in front of each new entry.

```
ACMEPACKET(realm-group)# destination-realm +Private
```

You can also remove single items in the list by using the minus sign (-) directly in front of the realm identifier.

```
ACMEPACKET(realm-group)# destination-realm -Private
```

8. **early-media-allow-direction**—Set the direction for which early media is allowed for this realm group. Valid values are:

- **none**—Turns off the feature for this realm group by blocking early media
- **reverse**—Allows early media to flow from called to caller
- **both (default)**—Allows early media to flow to/from called and caller

9. Save and activate your configuration.

SIP SDP Insertion for Re-INVITES

If your network contains some SIP endpoints that do not send SDP in ReINVITES but also contains others that refuse INVITES without SDP, this feature can facilitate communication between the two types. The Net-Net SBC can insert SDP into outgoing INVITE messages when the corresponding, incoming INVITE does not contain SDP.

You can also use this feature when the network devices used in H.323-SIP interworking do not include SDP in the INVITES sent to SIP endpoints. In this case, the Net-Net SBC can insert SDP in the outgoing INVITE messages it forwards to the next hop.

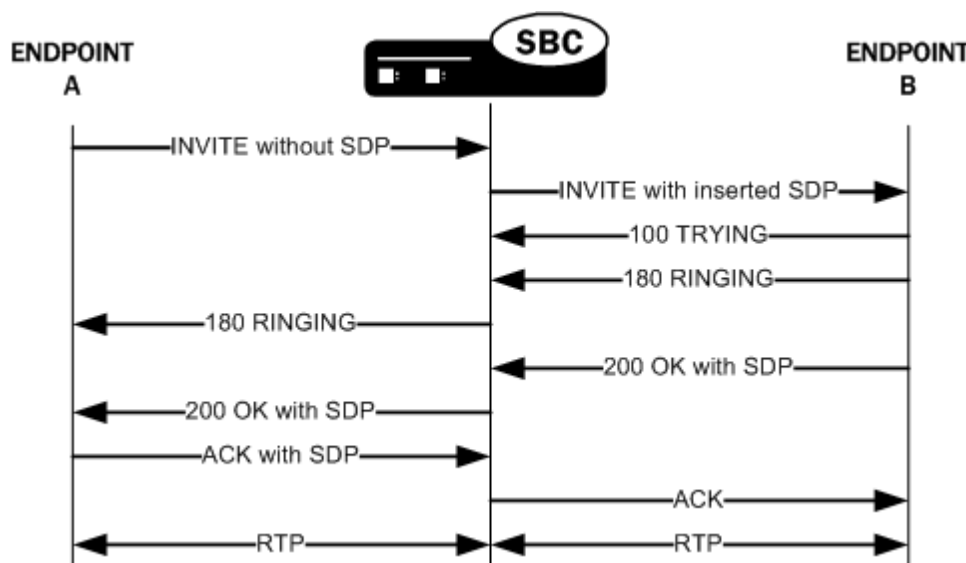
This feature works for both INVITES and ReINVITES.

How It Works

This section explains how the SDP insertion feature works for INVITES and ReINVITES. The examples used this section are both pure SIP calls. Even when you want to use this feature for IWF calls, though, you configure it for the SIP side.

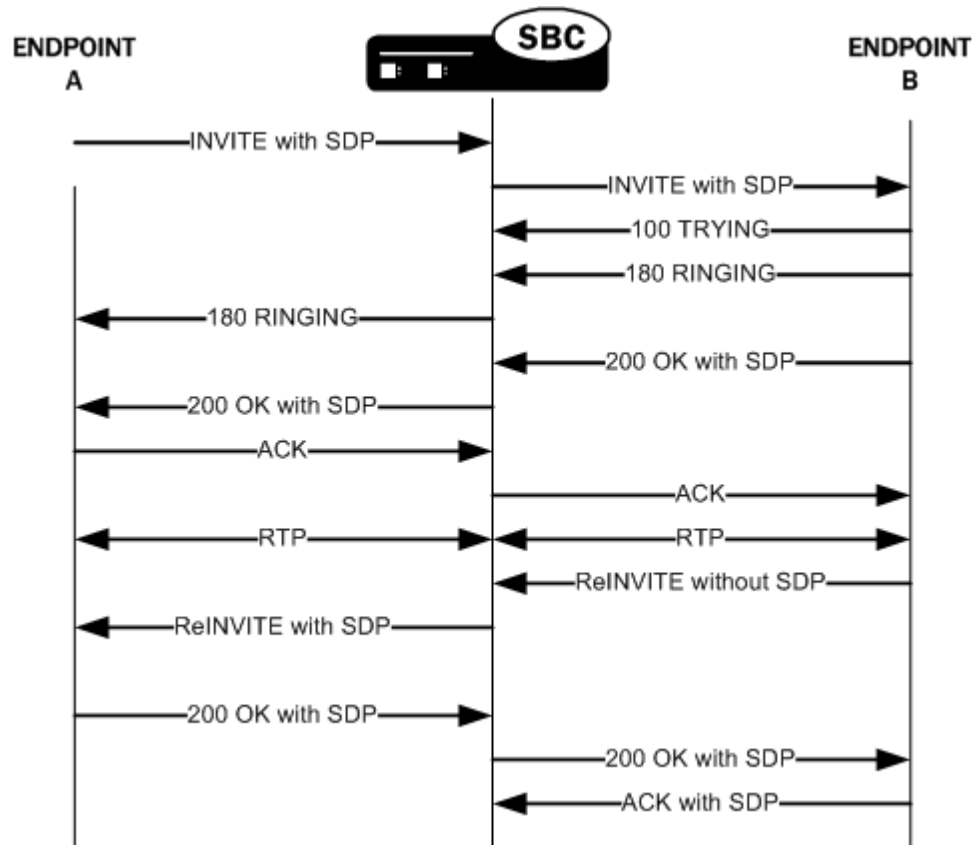
SDP Insertion for SIP INVITES

With the parameters mentioned above appropriately configured, the Net-Net SBC inserts SDP into an outgoing INVITE when the corresponding incoming INVITE has none. Because no SDP information is available for the session, the Net-Net SBC uses a media profile from a list of them you configure and then apply for SDP insertion.



SDP Insertion for SIP ReINVITES

The section explains SDP insertion for ReINVITES, using a case where SIP session has been established with an initial INVITE containing SDP. In the diagram below, you can see the initial INVITE results in a negotiated media stream. But after the media stream is established, Endpoint B sends a ReINVITE without SDP to the Net-Net SBC. In this case, the Net-Net SBC uses the negotiated media information from the initial INVITE to insert when the ReINVITE has no SDP. It then sends this ReINVITE with inserted SDP to the next hop signaling entity.



SDP Insertion for INVITEs and ReINVITEs

The Net-Net SBC can be configured to insert configured SDP into both INVITEs and ReINVITEs on a selected SIP interface by configuring the **add sdp invite** parameter to the value, **both**. In addition, when adding a SDP to a reINVITE, the Net-Net SBC never includes any a=inactive or a=sendonly SDP lines.

ACLI Instructions and Examples

This section shows you how to configure SDP insertion for the calls cases described above.

Configuring SDP Insertion for SIP INVITEs

To work properly, SDP insertion for SIP invites requires you to set a valid media profile configuration. For more information about how to set up media profiles, see the *Net-Net Configuration Guide*.

To enable SDP insertion for INVITEs:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-config)#
```

4. **add-sdp-invite**—Change this parameter from disabled (default), and set it to **invite**.
5. **add-sdp-profile**—Enter a list of one or more media profile configurations you want to use when the Net-Net SC inserts SDP into incoming INVITEs that have no SDP. The media profile contains media information the Net-Net SBC inserts in outgoing INVITE.

This parameter is empty by default.

6. Save and activate your configuration.

Configuring SDP Insertion for SIP ReINVITEs

In this scenario, the Net-Net SBC uses the media information negotiated early in the session to insert after it receives an incoming ReINVITE without SDP. The Net-Net SBC then sends the ReINVITE with inserted SDP to the next hop signaling entity. You do not need the media profiles setting for ReINVITEs.

To enable SDP insertion for ReINVITEs:

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

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)#
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-config)#
```

4. **add-sdp-invite**—Change this parameter from disabled (default), and set it to **reinvite**.

5. Save and activate your configuration.

Restricted Media Latching

This section explains how to configure restricted media latching, which lets the Net-Net SBC latch only to media from a known source IP address, in order to learn and latch the dynamic UDP port number. The restricting IP address's origin can be either the SDP information or the SIP message's Layer 3 (L3) IP address, depending on the configuration.

About Latching

Latching is when the Net-Net SBC listens for the first RTP packet from any source address/port for the destination address/port of the Net-Net SBC. The destination address/port is allocated dynamically and sent in the SDP. After it receives a RTP packet for that allocated destination address/port, the Net-Net SBC only allows subsequent RTP packets from that same source address/port for that particular Net-Net SBC destination address/port. Latching does not imply that the latched source address/port is used for the destination of the reverse direction RTP packet flow (it does not imply the Net-Net SBC will perform symmetric RTP).

Restricted Latching

The Net-Net SBC restricts latching of RTP/RTCP media for all calls within a realm. It latches to media based on one of the following:

- SDP: the IP address and address range based on the received SDP c= connect address line in the offer and answer.
- Layer 3: the IP address and address range based on the received L3 IP address of the offer or answer. This option is for access registered HNT endpoints. If the L3 IP address is locally known and cached by the Net-Net SBC as the public SIP contact address, that information could be used instead of waiting for a response. The Net-Net SBC might use the L3 IP address restriction method for all calls regardless of whether the endpoint is behind a NAT or not, for the same realms.

Symmetric Latching

A mode where a device's source address/ports for the RTP/RTCP it sends to the Net-Net SBC that are latched, are then used for the destination of RTP/RTCP sent to the device.

How it Works

After allocating the media session in SIP, the Net-Net SBC sets the restriction mode and the restriction mask for the calling side as well as for the called side. After receiving the calling SDP, the Net-Net SBC sets the source address (address and address prefix) in the appropriate flow (the flow going from calling side to the called side). After receiving the SDP from the called side, the Net-Net SBC sets the source address in the flow going from the called side to the calling side.

The Net-Net SBC uses either the address provided in the SDP or the layer 3 signaling address for latching. You also configure the Net-Net SBC to enable latching so that when it receives the source flow address, it sets the address and prefix in the NAT flow. When the NAT entry is installed, all the values are set correctly.

Enabling restricted latching may make the Net-Net SBC wait for a SIP/SDP response before latching, if the answerer is in a restricted latching realm. This is necessary because the Net-Net SBC does not usually know what to restrict latching to until the media endpoint is reached. The only exception could be when the endpoint's contact/IP is cached.

Relationship to Symmetric Latching

The current forced HNT symmetric latching feature lets the Net-Net SBC assume devices are behind NATs, regardless of their signaled IP/SIP/SDP layer addresses. The Net-Net SBC latches on any received RTP destined for the specific IP address/port of the Net-Net SBC for the call, and uses the latched source address/port for the reverse flow destination information.

If both restricted latching and symmetric latching are enabled, the Net-Net SBC only latches if the source matches the restriction, and the reverse flow will only go to the address/port latched to, and thus the reverse flow will only go to an address of the same restriction.

- Symmetric latching is enabled.

If symmetric latching is enabled, the Net-Net SBC sends the media in the opposite direction to the same IP and port, after it latches to the source address of the media packet.

- Symmetric latching is disabled.

If symmetric latching is disabled, the Net-Net SBC only latches the incoming source. The destination of the media in the reverse direction is controlled by the SDP address.

Example 1

A typical example is when the Net-Net SBC performs HNT and non-HNT registration access for endpoints. Possibly the SDP might not be correct, specifically if the device is behind a NAT. Therefore the Net-Net SBC needs to learn the address for which to restrict the media latching, based on the L3 IP address. If the endpoint is not behind a NAT, then the SDP could be used instead if preferred. However, one can make some assumptions that access-type cases will require registration caching, and the cached fixed contact (the public FW address) could be used instead of waiting for any SDP response.

Example 2

Another example is when a VoIP service is provided using symmetric-latching. A B2BUA/proxy sits between HNT endpoints and the Net-Net SBC, and calls do not appear to be behind NATs from the Net-Net SBC's perspective. The Net-Net SBC's primary role, other than securing softswitches and media gateways, is to provide symmetric latching so that HNT media will work from the endpoints.

To ensure the Net-Net SBC's latching mechanism is restricted to the media from the endpoints when the SIP Via and Contact headers are the B2BUA/proxy addresses and not the endpoints', the endpoint's real (public) IP address in the SDP of the offer/answer is used. The B2BUA/proxy corrects the c= line of SDP to that of the endpoints' public FW address.

The Net-Net SBC would then restrict the latching to the address in the SDP of the offer from the access realm (for inbound calls) or the SDP answer (for outbound calls).

ACLI Instructions and Examples

To configure restricted latching:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Select the realm where you want to apply this feature.


```
ACMEPACKET(realm-config)# select
identifier:
1: Acme_Realm <none>          0.0.0.0

selection:1
ACMEPACKET(realm-config)#
```
5. **restricted-latching**— Enter the restricted latching mode:
 - none—no latching used

- **sdp**—use the address provided in the SDP for latching
 - **peer-ip**—use the layer 3 signaling address for latching
6. **restriction-mask**— Enter the number of address bits you want used for the source latched address. If set to 32, the complete IP address is matched. This field will be used only if the restricted-latching is used.
 7. Save your work using the ACLI **done** and **save** commands.

Enhanced SIP Port Mapping

This section explains how to configure SIP port mapping feature to support:

- Anonymous requests from endpoints
- Cases where endpoints dynamically change transport protocols between UDP and TCP

Anonymous Requests

If a SIP endpoint sends an INVITE message with a From header that is anonymous, the Net-Net SBC can find the registration cache entry by using the Contact and Via headers. In cases such as instant messaging (IM), where there is no Contact header, the Net-Net SBC can use the Via header.

The Net-Net SBC's checks whether the **reg-via-key** option is configured for the access-side SIP interface where a REGISTER is received. If the option is enabled, the Net-Net SBC makes the via-key by adding the IP address from the Via header to the firewall address (if there is a firewall present between the Net-Net SBC and the endpoint).

When an INVITE arrives at a SIP interface where this option is enabled, the Net-Net SBC determines whether the From header is anonymous or not. If it is anonymous, then the Net-Net SBC uses the Via-key to find the registration entry.

ACLI Instructions and Examples

To enable support for anonymous SIP requests:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **options +reg-via-key** and press <Enter>.


```
ACMEPACKET(sip-interface)# options +reg-via-key
```

If you type **options reg-via-key** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.
5. Save and activate your configuration.

SIP Registration Via Proxy

The Net-Net SBC supports a number of features that require it to cache registration information for UAs (endpoints) registering and receiving requests through it. For those features to operate correctly, the Net-Net SBC must act as the outbound proxy through which these endpoints register.

In order to support deployments where a proxy sits between the Net-Net SBC and the endpoints, the Net-Net SBC must consider the bottom Via header it receives from endpoints when constructing and matching cache registration entries. And when you use SIP port mapping, the system must use the bottom Via header as a way to determine the endpoint uniquely so that it can have a unique mapping port when the SIP interface is configured with the `reg-via-key=all` option.

Using the `reg-via-proxy` option, you can enable your Net-Net SBC to support endpoints that register using an intervening proxy. You can set this option either for a realm or for a SIP interface. If you use it for a SIP interface, add to the SIP interface pointing toward the proxy and endpoints—the access side.

Considerations for Reg-Via-Key and Port Mapping

When you set the **reg-via-proxy** option, the Net-Net SBC includes the bottom Via header from received requests in the registration cache Via Key. The system also uses it for determining whether or not the request matches a registration cache entry. Each unique bottom Via received a unique mapping port when you turn SIP port mapping on and set the SIP interface with the **reg-via-key=all** option.

Request Routing

So that requests addressed to the corresponding registered contact are routed to the proxy, the Net-Net SBC includes the intervening proxy (i.e., the top Via) in the routing information for the registration cache when you set **reg-via-proxy**. To carry out this routing scheme, the system adds a Path header (if none is present) to the REGISTER. But it removes the Path header prior to sending the REGISTER to the registrar.

Note that when the received REGISTER contains a Path header, the Net-Net SBC uses it for routing requests toward the endpoint and includes it in the forwarded REGISTER request—as is the case when you do not enable SIP registration via proxy.

ACLI Instructions and Examples

To configure SIP registration via proxy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **options +reg-via-proxy** and press <Enter>.


```
ACMEPACKET(sip-interface)# options +reg-via-proxy
```

If you type **options reg-via-proxy** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

5. Save and activate your configuration.

SIP Registration Cache Limiting

Using SIP registration cache limiting for SIP endpoint access deployments, you can restrict the size of the SIP registration cache for the global SIP configuration.

You can implement this feature if you have been seeing issues where, either due to network failure scenarios or incorrect sizing of system capabilities, the Net-Net SBC and/or the SIP registrar cannot support the number of registering endpoints. Although the Net-Net SBC protects itself and the registrar against SIP REGISTER floods, conditions can still occur where too many legitimate endpoints attempt to register with the registrar via the Net-Net SBC.

By enabling SIP registration cache limiting, you restrict the number of legitimate endpoints that can register. The Net-Net SBC rejects any endpoints beyond the limit you set. If you do not want to use this feature, simply leave the `reg-cache-limit` parameter set to its default of 0, meaning there is no limit to the entries in the SIP registration cache.

How It Works

When you limit the number of registered endpoints allowed in the Net-Net SBC's registration cache, the Net-Net SBC analyzes each registration before starting to process it. First, the Net-Net SBC checks the contact header to determine if it is already in the list of contacts for the user. If it finds the contact in its cache list, the Net-Net SBC treats the registration as a refresh; it treats any other headers as new. Note that the Net-Net SBC checks the message prior to making any changes to the cache because it must either accept or reject the message as a whole.

The Net-Net SBC adds the number of new contacts to the number already present in the cache, and rejects any registration with a contact that would cause it to exceed its limit. Rejection causes the Net-Net SBC to send a response communicating that its registration cache is full. The default response is the **503 Registration DB-Full** message, but you can use the SIP response mapping feature to use another message if required.

You can set an option in the global SIP configuration that defines the value in the `Retry-After` header. The Net-Net SBC sends this header as part of its rejection response when the registration cache is full. Another option sets the percentage of the registration cache size which, if exceeded, causes the Net-Net SBC to send an alarm.

About Registration Cache Additions, Modifications, and Removals

When it receives a REGISTER message with new contact information for a user, the Net-Net SBC considers it an addition to the cache and augments the number of registration cache entries. Then the Net-Net SBC forwards the message to the registrar, and—when and only when the registrar returns both the original and new contacts in the 200 OK—the registration cache count stays the same. However, if the registrar returns only the new contact (making this a case of modifications), then the

Net-Net SBC removes the old contact information and subtracts accordingly from the number of registration cache entries.

Thus the Net-Net SBC does not know whether a REGISTER might result in an addition or a modification until it receives a response from the registrar. For this reason, the Net-Net SBC first assumes it is to make an addition, and then updates the registration cache and count when it has the necessary information from the registrar.

The registration cache count does not reflect removals during the rejection check because the Net-Net SBC ignores registration messages or expires headers with their expires values set to zero when it counts new entries. The fact that removals take place after additions and modifications means that messages which remove one contact while adding another might be rejected. That is, the addition might exceed the registration cache limit before any removal can take place to make room for it.

Registration Cache Alarm Threshold

A percentage of the registration cache limit, the registration cache alarm threshold is a configurable value you can set to trigger an alarm when the registration cache is reaching its limit. When exceeded, this threshold triggers the generation of an alarm and SNMP trap. When registrations fall back beneath the threshold, the Net-Net SBC clears the alarm and sends a clear trap.

This alarm is Major in severity, and its text reads as follows:

```
Number of contacts <registration count> has exceeded the registration
cache threshold <threshold %> of <registration cache limit value>.
```

Notes on Surrogate Registration

The Net-Net SBC does not, under any circumstances, reject surrogate registrations on the basis of the registration cache limit. However, surrogate registrations generate contacts, and so they do add to the global registration count. In the case where the surrogate registrations add to the registration count to the extent the count exceeds the limit you configure, you will have more registrations in the cache than the configured limit.

SIP Registration Event Package Support

Certain endpoints subscribe to the Registration Event Package, RFC 3680, which defines how SIP user agents can request and obtain notifications about registration events. Previously, the Net-Net SBC passed the Subscribe and Notify messages of this package transparently, without modifying the XML bodies of either. However, in many cases the XML body can contain IP addresses, contact URIs, and expires times that the Net-Net SBC needs to modify for proper operation. This new feature enables the Net-Net SBC to modify correctly the XML body for the Registration Event Package.

In addition to resolving this type of issue, enabling registration event package support on your system provides the functions described below:

- The Net-Net SBC performs NAT on all contacts in the **reginfo**, regardless of their state.
- The Net-Net SBC performs NAT on the address of record (AoR) attribute of the Registration element when it matches an existing cache entry. When either the Contact-URI or the AoR does not match a cache entry and the host part of the URI is an IP address, the Net-Net SBC will NAT the host part using the applicable SIP NAT configuration

- Contacts are found in the XML URI element for the contact. But if there is no URI element, then the Net-Net SBC uses the Contact element information for the contact.
- If the “expires” attribute in the Contact element is a value other than zero, the Net-Net SBC uses (inserts) the expires values from the registration cache.
- This feature also introduces delayed deletion from the registry cache. When a 200 OK comes back in response to a REGISTER message and the 200 OK does not include all previously registered contacts, the missing contacts are deleted. If the global SIP configuration option `contact_cache_linger=XX` (where XX is the number of seconds to wait before deleting), then the contacts to be deleted remain for the specified number of seconds before they in fact are deleted.

Updating Expiration Values

This feature also supports updating the expiration values for the registration cache when a Contact element has the expires attribute. For this support, the following apply:

- If the value of the expires attribute is greater than the expiration value for the access-side registration cache entry, the Net-Net SBC replaces the XML expires attribute value with the cached one from the access side.
- If the value of the XML expires attribute is less than the core-side expiration value for the core-side registration cache entry, the Net-Net SBC updates the core-side expiration value with the value from the expires attribute. Further, the Net-Net SBC adjusts the access-side expiration value of the registration cache in these ways:
 - If the value of the XML expires attribute is less than the current access-side expiration value for the registration cache entry, the Net-Net SBC sets the access-side expiration value to be equal to the value in the expires attribute.
 - Otherwise, the Net-Net SBC leaves the expires value for the access-side expiration value for the registration cache entry unchanged. If this happens, the Net-Net SBC replaces the value of the XML expires attribute with the adjusted access-side expiration value.
- If the expires attribute from a Contact element is 0 (meaning that the core is removing the registration), the Net-Net SBC removes that Contact-URI from its registration cache. And if the registration cache entry has no remaining Contact-URIs, the Net-Net SBC deletes the registration cache entry altogether.

ACLI Instructions and Examples

You enable this feature as part of the global SIP configuration, using that configuration’s **options** parameter. You can set the parameter for the registration event package support, and optionally configure the number of seconds you want to keep a contact in the registration cache before it is deleted. These are the two options:

- **nat_reginfo**—Enables registration event package support
- **contact_cache_linger=XX**—Number of seconds to wait before a contact is deleted from the cache (where XX is the number of seconds)

To enable SIP Registration overload protection on your Net-Net SBC:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

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

```
ACMEPACKET(session-router)# sip-config
```

```
ACMEPACKET(sip-config)#
```

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**nat_reginfo**), and then press <Enter>. Follow the same steps to add the **contact_cache_linger=XX** (where XX is the number of seconds to keep a contact in the cache before deleting it) option.

```
ACMEPACKET(sip-config)# options +nat_reginfo
```

```
ACMEPACKET(sip-config)# options +contact_cache_linger=5
```

If you type either of these options without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

5. Save and activate your configuration.

SIP Method-Transaction Statistic Enhancements

In prior releases, the Net-Net SBC tracks SIP session agents, SIP interfaces and SIP realms on a global level. Only counters that are related to session rates and constraints are displayed.

How It Works

You can now enable your Net-Net SBC to track transaction messages for specific SIP session agents, SIP realms, and SIP interfaces.

The following SIP methods are tracked for Recent, Total, and Period Max values:

- ACK
- BYE
- CANCEL
- INFO
- INVITE
- MESSAGE
- NOTIFY
- OPTIONS
- PRACK
- PUBLISH
- REFER
- REGISTER
- SUBSCRIBE
- UPDATE
- “other” (unknown)

With this tracking enhancement, the **show sip** command has been updated with a new “method” argument which allows you to query statistics for a particular method

for a given SIP agent, SIP interface, or SIP realm. You must first enable the extra method statistics parameter in the SIP configuration.

For this feature, you use a configuration called an enforcement profile that allows you to configure sets of SIP methods that you want applied to: the global SIP configuration, a SIP interface, a realm, or a SIP session agent. The enforcement profile is a named list of allowed methods that you configure and then reference from the configuration where you want those methods applied.

Note: This detailed logging feature should be set to disabled if unneeded due to it drawing system resources away from call processing.

ACLI Instructions and Examples

This section explains how to configure the expanded SIP method statistics tracking and how to view method statistics for a particular SIP agent, interface, or realm.

Enabling the SIP Method Tracking Enhancements

You must first enable the extra method statistics in the global SIP configuration.

To enable or disable the expanded SIP Method statistics tracking:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
```
4. **extra-method-stats**—If you want to use the expanded SIP Method tracking feature, change this parameter from **disabled** (default) to **enabled**.

Save and activate your configuration.

National Security and Emergency Preparedness for SIP

The Net-Net SBC supports Emergency Telecommunications Service (ETS), which gives priority treatment of National Security and Emergency Preparedness (NSEP) communications for IP network infrastructures. ETS can increase the likelihood that calls, sessions, and other communications will be successfully completed when they are initiated by government-authorized users over the public network infrastructure. Legacy circuit-switched services such as Government Emergency Telecommunications Service (GETS) and Wireless Priority Service (WPS) also fall under the ETS rubric, and are now also supported on the Net-Net SBC.

To provide this support, you can enable the Net-Net SBC to act on SIP calls that contain an ETS dial number (DN) and/or the SIP Resource-Priority header that carries ETS resource values.

How It Works

The Net-Net SBC identifies ETS calls by using the system's pre-existing network management controls (NMC) functionality. With NMC and Resource-Priority header (RPH) support enabled on your system, the Net-Net SBC detects ETS calls and provides the appropriate treatment for them.

The Net-Net SBC supports this feature by treating ETS calls based on the r-value parameter in the Resource-Priority header. The r-value is a key piece of information because it defines the resource priority that the call originator requests. The r-value parameter provides namespaces and priorities that the Net-Net SBC can manipulate in outgoing traffic.

In addition to a new RPH profile configuration containing information about how to treat RPHs, new parameters in the global SIP configuration and NMC configuration have been added. The RPH profile is applied to an NMC rule, where they determine r-values, a media policy to use, and what type of call treatment to apply. Also applies to an NMC rule, the new RPH policy configuration provides information about which r-values to insert and which to override.

Licensing

To enable NSEP for SIP on your Net-Net SBC, you must obtain and install a new license. If properly installed on your system, it appears as NSEP RPH in the display issued when you use the ACLI **show** command in the license configuration.

For information about how to obtain an NSEP RPH license, contact your Acme Packet sales representative.

Matching by NMC and by RPH

When a Net-Net SBC has been enabled to act on RPH, it checks incoming requests for RPH, tries to parse that RPH, and then rejects requests in the circumstances listed below. For all of these rejections, the Net-Net SBC logs the error at the TRACE level.

- Request with multiple instances of the same namespace in the RPH—The Net-Net SBC sends out a **400 Bad Request** response with the “Invalid RPH - Namespace repeated” header showing that there are multiple instances of the same namespace in the RPH.
- Request with invalid resource priority for a namespace—The Net-Net SBC sends out a **400 Bad Request** response with the “Invalid RPH - Invalid rvalue: x” showing that there is an invalid resource value (where x is the invalid value).
- Request with WPS namespace, but without ETS namespace—The Net-Net SBC sends out a **400 Bad Request** response with the “Invalid RPH - No ETS value” header showing that there is no ETS namespace.

If the Net-Net SBC successfully parses the RPH, it identifies the ETS call by checking the Request-URI of the incoming request against destination identifiers that you configure in the NMC rules. If there is a match between the request’s ETS DN and the destination value identifier in the NMC rules, the Net-Net SBC tags the call; note that NMC rules need to be configured with the **rph-feature** parameter set to enabled to identify an ETS call properly. If there is no match to an NMC rule, then the Net-Net SBC performs matching based on RPH by comparing resource values (r-values) in the RPH with values you set in the RPH profile configuration.

For an ETS call that matches by ETS DN and NMC rule, the Net-Net SBC checks the NMC rule to determine if it has an RPH profile (with r-values) assigned to it. If so, the Net-Net SBC continues by comparing the RPH profile’s r-values against those in the request’s RPH. In cases where the RPH does not contain a recognized value r-value, the Net-Net SBC:

- Processes the call as it normally would (as a non-ETS call) without changing the RPH if the resource-priority option tag is not present in the Required header (for an INVITE only and not any other requests or response from which RPH would be deleted)

- Rejects the Request when the Require header has the resource-priority header; or, inserts an Accept-Resource-Priority header (ARPH) in the response if the **insert-arp-header** parameter option is enabled

However, the call goes through the Net-Net SBC as an ETS call when it is matched by ETS DN and the applicable NMC does not have an RPH profile assigned. According to the settings in the NMC rule, the Net-Net SBC either diverts or rejects such a call. And when the call matches by RPH rather than ETS DN, the Net-Net SBC applies the configured RPH profile from the relevant NMC rule.

It can be the case that non-ETS calls have RPH in their requests. Here, the Net-Net SBC call treatment is performed according to the settings in the matching RPH profile when there is no matching NMC rule. When you configure treatment as “reject,” then the Net-Net SBC rejects the call with a **417 Unknown-Resource Priority** status code. When you set the treatment to either “accept” or “priority,” the Net-Net SBC allows the call to proceed as a non-ETS call or as a priority call.

The ETS r-value can appear in ACK, BYE, INFO, PRACK, REFER and UPDATE requests. In cases when it does and the session with which the request is associated is a non-ETS call, the Net-Net SBC removes the RPH from the request before forwarding it and logs a TRACE-level error. The Net-Net SBC also removes RPH from responses before forwarding them and logs a TRACE-level error when responses contain RPH headers with ETS values for non-ETS sessions.

Call Treatment

This section describes how ETS calls are treated as they traverse the Net-Net SBC.

Call Treatment	Description
Routing	ETS calls are routed the same way as any other calls are, except when the applicable NMC rule's treatment type is “divert,” and rule defines the next hop. This route takes precedence over other normal routes.
Local NMC	ETS calls are exempt from the local NMC, including: session agent constraints, bandwidth constraints (e.g., per-realm bandwidth), per-user CAC, and CPU constraints. However, the call is subject to the ETS congestions control threshold. Licensing session constraints apply.
ETS Call Congestion Control	ETS calls are subject to congestion control constraints that you configure specifically for this type of traffic. In the global SIP configuration, you set up one option that defines a load limit (greater than that set for normal calls).
ETS CAC	Although the Net-Net SBC uses the call rate control value in the applicable NMC rule, you can also enforce call rate on a per-user basis for ETS calls.

Generating Egress RPH

For each ETS call, the Net-Net SBC generates RPH for the outgoing request. It forms this RPH according to the information in the NMC rule. The outgoing request types are INVITE, ACL, BYE, CANCEL, INFO, PRACK, REFER, and UPDATE.

Request RPH Status	Generated Egress RPH
Incoming request without RPH (matched by ETS DN)	Outgoing RPH value becomes the r-value set in the Insert-r-value parameter in the RPH policy applied to the NMC rule.

Request RPH Status	Generated Egress RPH
Incoming request without RPH (matched by ETS DN)	If the insert-r-value parameter is empty in the RPH policy applied to the NMC rule or there is no RPH policy applied to the NMC rule, then the egress RPH will also not have RPH.
Incoming request has RPH	Egress RPH is the same as the ingress if the NMC rule has an RPH policy applied but the override-r-value for the policy is empty or if there is not RPH policy applied to the NMC rule. If the override-r-value for the policy is set, then the egress RPH is set to that value.

For example, given an incoming request with the resource priority `ets.0`, `dsn.flash` and an RPH policy with an override value of `wps.1`, `ets.1`, the egress request would be sent with a resource-priority of `wps.1`, `ets.1`, `dsn.flash`.

The Net-Net SBC also includes RPH in the following series of responses, even when the downstream SIP entity does not respond with an RPH: 1xx, 2xx, 3xx, 4xx, 5xx, and 6xx. The **401 Unauthorized** response is an exception.

Media Treatment

If the RPH profile set in an NMC names a media policy, then the Net-Net SBC implements it for the ETS call. This media policy overrides any media policy set in the realm configuration.

The possible Differentiated Services Code Point (DSCP) values for an ETS call are:

- Audio—Applied to the respective media for an ETS call
- Video—Applied to the respective media for an ETS call
- SIP—Applied to the ETS calls' SIP signaling messages, only for the egress call leg for the ETS session

ACLI Instructions and Examples

This section shows you how to configure RPH profiles and policies that enable the Net-Net SBC to act on SIP calls that have an ETS DN and/or an RPH carrying ETS resources values. There are also settings for the global SIP configuration and for the NMC rule configuration that support this feature.

In addition, note that:

- You must set a media policy for the RPH profile to use. Check your system configuration and note the name of the media policy that best suits your needs.
- Valid values for the parameters that take r-values are `wps.x` and `ets.x`, where `x` is 0 through 4.

Remember to save and activate your configuration after you have completed the processes detailed in this section.

Setting Up and Applying RPH Policy

The RPH policy is a configuration on the Net-Net SBC that you apply to NMC rules. It designates the following for ETS/WPS namespaces:

- An override resource value—Resource value used to override the incoming RPH's resource value
- An insert resource value—Resource value inserted when the Net-Net SBC does not recognize the RPH or the incoming request has no RPH

Note that RPH policies do not apply for DSN, DRSN, Q.735, or any other type of namespace; these remain untouched in outgoing requests.

To configure an RPH policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **rph-policy** and press <Enter>. From here, you can configure the individual parameters for the RPH policy.


```
ACMEPACKET(session-router)# rph-policy
ACMEPACKET(rph-policy)#
```
4. **name**—Enter the name that uniquely identifies this RPH policy. This is the value you use to apply the policy in the NMC rules configuration. There is no default for this parameter, and you are required to set it.
5. **override-r-value**—Enter the value that the Net-Net SBC uses to override r-values in the original RPH. Excluding keywords **add** and **delete** when a list is already configured replaces the entire list.


```
ACMEPACKET(rph-policy)# override-r-value ets.1
```
6. **insert-r-value**—Enter the value that the Net-Net SBC inserts into the RPH. Excluding keywords **add** and **delete** when a list is already configured replaces the entire list.


```
ACMEPACKET(rph-policy)# insert-r-value wps.1
```
7. Save your work using the ACLI **done** command.
8. Save and activate your configuration.

To apply an RPH policy to an NMC rule:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
ACMEPACKET(net-management-control)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
4. **rph-policy**—Enter the name of the RPH policy that you want to apply for this NMC rule. This parameter is empty by default; if you do not set an RPH policy, none will be applied.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Setting Up and Applying RPH Profile

The RPH profile contains information about how the Net-Net SBC should act on the namespace(s) present in a Resource-Priority header (if any). The list of resource values in this configuration calls out the resource values (or r-values) recognizable to the Net-Net SBC; the ETS and WPS namespaces are supported.

You also set a media policy for the RPH profile to use; it defines the Differentiated Services Code Point (DSCP) that the Net-Net SBC uses for media or signaling packets belonging to the egress call leg for the ETS session.

The call treatment parameter tells the Net-Net SBC what to do with a non-ETS call that has RPH in its request; the call can be allowed, rejected, or treated as a priority call.

To configure an RPH profile:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **rph-profile** and press <Enter>. From here, you can configure the individual parameters for the RPH policy.


```
ACMEPACKET(session-router)# rph-profile
ACMEPACKET(rph-profile)#
```
4. **name**—Enter the name that uniquely identifies this RPH profile. This is the value you use to apply the profile in the NMC rules configuration. There is no default for this parameter, and you are required to set it.
5. **r-values**—Enter one or more r-values that the Net-Net SBC is to recognize for matching purposes. When you enter more than one value in the list, you type the name of the parameter followed by a <Space>, open quotation mark, the values for the list separated by spaces, a closed quotation mark. Then press <Enter>. Excluding keywords **add** and **delete** when a list is already configured replaces the entire list.

You must enter them in the order reflected below (a WPS and then an ETS value). A WPS call always has to have an ETS namespace.

```
ACMEPACKET(rph-profile)# r-values "wps.0 ets.2"
```
6. **media-policy**—Enter the name of a media policy configuration that you want applied for this RPH profile. The Net-Net SBC implements this media policy for the ETS call, and this media policy overrides any media policy set in the realm configuration.
7. **call-treatment**—Enter the call treatment method for a non-ETS call that contains RPH matching it to this profile. The valid values are: **accept** (default, the call proceeds as it normally would), **reject** (the Net-Net SBC rejects the call with the 417 Unknown-Resource Priority status code), and **priority** (Net-Net SBC treats the call as a priority call).
8. Save your work using the ACLI **done** command.
9. Save and activate your configuration.

To apply an RPH profile to an NMC rule:

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


```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
ACMEPACKET(net-management-control)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
4. **rph-profile**—Enter the name of the RPH profile that you want to apply for this NMC rule. This parameter is empty by default; if you do not set an RPH profile, none will be applied.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Enabling NSEP for an NMC Rule

In addition to the RPH policy and RPH profile you can set for an NMC rule, you also need to set the state of this feature for the NMC rule.

To enable NSEP for an NMC rule:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
ACMEPACKET(net-management-control)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
4. **rph-feature**—Enter **enabled** if you want to turn the NSEP feature on for this NMC rule. This parameter is set to **disabled** by default.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Global SIP Configuration Settings: Enabling NSEP

For the global SIP configuration, you can turn the NSEP feature on, and you can also set parameters that support call admission and congestion control.

In addition, you can enable the insertion of the ARPH header in a response when the resource-priority tag is present in the Require header and the Net-Net SBC rejects the request with a 417 **Unknown Resource-Priority** response. The ARPH value is the list of r-values you set in the RPH profile.

To enable NSEP for the global SIP configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-config** and press <Enter>.

```
ACMEPACKET(session-router)# sip-config
```

```
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.

4. **rph-feature**—Enter **enabled** if you want to turn the NSEP feature on for the global SIP configuration. This parameter is set to **disabled** by default.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Global SIP Configuration Settings: Enabling CAC and Congestion Control

To set call admission and congestion control parameters for NSEP:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-config** and press <Enter>.

```
ACMEPACKET(session-router)# sip-config
```

```
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.

4. **nsep-user-sessions-rate**—Enter the maximum INVITEs per second to admit for ETS calls on a per-user basis. To enable NSEP call admission control (CAC), you must change the parameter value from **0**; if you leave this parameter set to **0**, then it is the same as disabling CAC for ETS calls.
5. **options**—To enable congestion control for ETS calls, you configure an option that sets the CPU threshold. If this threshold is exceeded, the Net-Net SBC rejects new ETS calls with the **503 Service Unavailable** response. The value you set here should be larger than the load limit value for normal calls; ETS calls are allowed even when the load limit threshold for normal calls is exceeded.

The threshold value can be between 0 and 100. Using a value of 0 or 100 for this parameter disables ETS call congestion control.

Set the options parameter by typing **options**, a <Space>, the option name **nsep-load-limit** with a “plus” sign in front of it, then the equal sign and the ETS call threshold you want to set. Then press <Enter>.

```
ACMEPACKET(sip-config)# options +nsep-load-limit=50
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save your work using the ACLI **done** command.
7. Save and activate your configuration.

Global SIP Configuration Settings: Enabling ARPH Insertion

To enable ARPH insertion in responses:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.
4. **options**—To enable ARPH insertion in responses type **options**, a <Space>, the option name **insert-arp-header** with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(sip-config)# options +insert-arp-header
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Debugging

The **show sip nsep** command is used to display the statistics collected for NS/EP calls. You can view statistics for all NS/EP calls on the Net-Net SBC either globally, or for those with a specific r-value. The command is entered as:

```
show sip nsep [<rvalue> | all]
```

For example:

```
ACMEPACKET# show sip nsep
19:10:18-153 sip1s@0.3.0
Session Stats
----- Recent ----- Lifetime -----
Active High Total Total PerMax High
Inbound Sessions - - 0 0 0 0
ACMEPACKET# show sip nsep ets.1
19:10:30-165 sip1s@0.3.0
Rvalue = ets.1
----- Recent ----- Lifetime -----
Active High Total Total PerMax High
Incoming Sessions 0 0 0 0 0 0
Outgoing Sessions 0 0 0 0 0 0
InbSessions Rej - - 0 0 0
OutbSessions Rej - - 0 0 0
```

The **show sip nsep all** command displays statistics for all configured r-values.

Resetting NSEP Statistics

To reset all NSEP statistics, you must perform the global **reset sip** command.

NSEP Enhancements

This group of NSEP enhancements covers:

- The ability to configure and monitor NSEP call admission control (CAC) as applied to session agents
- Call treatment when the Net-Net SBC receives a SIP INVITE with an RPH matching the network management controls (NMC) with an ETS DN, but whose r-values do not match the NMC's rph-profile
- Log level changes for instances when users or session agents exceed constraints

NSEP for Session Agents

In earlier releases, the Net-Net SBC supports NSEP-related CAC only for users and for NMC. You can now configure a sessions-per-second rate for session agents. Set in the global SIP configuration, this rate applies to all SIP session agents. When a session agent exceeds the limit, the Net-Net SBC rejects it with a **503 Service Unavailable** message.

ACLI Instructions and Examples

To configure NSEP limits for SIP session agents:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **nsep-sa-sessions-rate**—Enter maximum acceptable number of SIP INVITES (NSEP sessions) per second to allow for SIP session agents. This parameter defaults to 0, meaning there is no limit.
5. Save and activate your configuration.

Viewing NSEP Burst Statistics for SIP Session Agents

The ACLI **show sipd** command now supports an **sa-nsep-burst** argument that displays the NSEP burst rate for all SIP session agents.

```
ACMEPACKET# show sipd sa-nsep-burst
```

Agent	Current Rate	Lifetime High
192.168.1.139	0	0
192.168.1.6	0	0
192.168.200.135	4	10

Call Treatment

When the Net-Net SBC receives a SIP INVITE with an RPH matching an NMC with an ETS DN, but whose r-values do not match the NMC's rph-profile, the Net-Net SBC behaves as follows:

- If the INVITE does not have the **resource-priority** option tag and:
 - If the matching NMS is set to **PRIORITY**, the call will be treated as an NSEP call. If some stored rph-profile match the r-value (not necessarily the one in

the NMC), the Net-Net SBC uses the media-policy from that rph-profile for the call. The rph-policy from the NMC (if present) also applies to the call.

- If the matching NMC is not set to PRIORITY, the Net-Net SBC will treat the call as a normal one.
- If the INVITE contains the `resource-priority` option tag, the Net-Net SBC will reject the call with the `417 Unknown Resource-Priority` message.

Log Level Changes

When the `nsep-user-sessions-rate` or `nsep-sa-sessions-rate` parameters have been exceeded, the Net-Net SBC will log a message to `log.sipd` if the log level is as low as `NOTICE`.

Because the log level for these messages is now `NOTICE`, the Net-Net SBC must guard against instances such as an NSEP attack that would fill the log file. Therefore, only ten such log messages will be logged for a 200-second windows—ten for a session agent and ten for a user, with a maximum of twenty if they fall in the same window.

SIP TCP Connection Reuse

You can configure your Net-Net SBC to reuse TCP connections created by SIP peering devices for outgoing SIP in-dialog and out-of-dialog request transactions.

The SIP draft `draft-ietf-sip-connect-reuse-07.txt` describes a way for SIP UAs to reuse connections created by a remote endpoint for outgoing requests for TLS. The Net-Net SBC does not support the model connection reuse signaled by a parameter; rather, it is provisioned on a per-session-agent basis.

How It Works

You enable SIP TCP connection reuse on a per-session-agent basis. The Net-Net SBC checks incoming TCP connection request to determine if they are from a session agent that has this feature turned on. When it is, the Net-Net SBC adds the connection's source address to its list of alias connections. This is a list of connections that the Net-Net SBC can use for outgoing requests rather than creating its own connection (as it does when this feature is not enabled). So if a preferred connection fails, the Net-Net SBC can refer to this list and use the alias connection.

The presence of an "alias" parameter in the Via header is just one mechanism that will call the Net-Net SBC to use the inbound TCP/TLS connection for outbound requests. The Net-Net SBC will automatically add an "alias" for the inbound connections in the following circumstances:

- The other end of the connection is behind a NAT. When the Net-Net SBC sees that the Via sent-by does not match the source address of the connection, it will automatically reuse the connection to deliver requests to the UA.
- The Contact address of a REGISTER request received on a TCP connection matches the source address and port. This is because the contact address is the ephemeral port the UA used to form the connection to the Net-Net SBC and, therefore, will not be listening on that port for inbound connections.

The presence of "reuse-connections" in the options field of the sip-interface will cause the Net-Net SBC to reuse all inbound TCP connections for sending requests to the connected UA.

ACLI Instructions and Examples

This section describes how to enable SIP TCP connection reuse for a session agent. Currently there are two options for the new **reuse-connections** parameter: **none** (which turns the feature off) and **tcp** (which enables the feature for TCP connections). You also set the re-connection interval.

To enable SIP TCP connection reuse for a session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.
4. **reuse-connections**—Enables or disables SIP TCP connection reuse. The default is **none**, which disables the feature. The valid values are:
 - **tcp** | **none**
5. **tcp-reconn-interval**—Enter the amount of time in seconds before retrying a TCP connection. The default for this parameter is **0**. The valid range is:
 - Minimum—0, 2
 - Maximum—300
6. Save and activate your configuration.

SIP Enforcement Profile and Allowed Methods

An enforcement profile allows you to configure sets of allowed SIP methods that are applied to the following configuration elements: the global SIP configuration, SIP interfaces, realms, SIP session agents. The enforcement profile itself is a list of allowed methods that you configure and then reference from the configuration where you want those methods applied.

In addition to the standard set of methods (e.g., INVITE, UPDATE, REGISTER), any other text you enter for the allowed-methods parameter becomes a new method that the Net-Net SBC will not block. You must ensure that you type the standard methods correctly without any typos or else you could accidentally block several messages.

Note: ACK, BYE, and CANCEL messages are implicitly not blocked.

Setting Up an Enforcement Profile

You need configure the enforcement profile with a name and the list of SIP methods you want to designate as allowed. Then you need to configure the global SIP configuration, a SIP interface, a realm, or SIP session agent to use the set.

To set up an enforcement profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **enforcement-profile** and press <Enter>.


```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```
4. **name**—Enter the name for the enforcement profile. This parameter has no default, but you must note it so that you can apply this set of allowed SIP headers in: the global SIP configuration, a SIP interface, a realm, or SIP session agent.


```
ACMEPACKET(enforcement-profile)# name EnfProfile1
```
5. **allowed-methods**—Enter a list of SIP methods that you want to allow for this set. Valid entries come from this list: INVITE, REGISTER, PRACK, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, UPDATE, MESSAGE, PUBLISH, *unknown*.

To enter multiple methods for the list, type the parameter name followed by a space, then the names of all methods you want to include each separated by a only a space.

```
ACMEPACKET(enforcement-profile)# allowed-methods INVITE REGISTER PRACK
```
6. Save and activate your configuration.

Applying an Enforcement Profile

You can apply an enforcement profile to: the global SIP configuration, a SIP interface, a realm, or SIP session agent. This section shows you how to do all four. Remember that if you are adding this functionality to a pre-existing configuration, you need to select the configuration you want to edit.

To apply an enforcement profile to the global SIP configuration:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to the global SIP configuration.
5. Save and activate your configuration.

To apply an enforcement profile to a SIP interface:

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


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

2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this SIP interface.
5. Save and activate your configuration.

To apply an enforcement profile to a SIP session agent:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this session agent.
5. Save and activate your configuration.

To apply an enforcement profile to a realm:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this realm.
5. Save and activate your configuration.

Monitoring

To view statistics for a particular SIP agent method:

1. From User or Superuser mode, type **show sip agent <name> <method>** and press <Enter>.


```
ACMEPACKET# show sip agents <agent name> <method>
```

To view statistics for a particular SIP interface method:

1. From User or Superuser mode, type **show sip interface <name> <method>** and press <Enter>.

ACMEPACKET# **show sip interface <interface name> <method>**

To view statistics for a particular SIP realm method:

1. From User or Superuser mode, type **show sip realm <name> <method>** and press <Enter>.

ACMEPACKET# **show sip realms <realm name> <method>**

Local Policy Session Agent Matching for SIP

When you enable the local policy session agent matching option in your global SIP configuration, you change the way local policies match session agents. Normally, the Net-Net SBC looks up and stores matched session agents configured as next hops so it does not need to perform the lookup while processing requests. In this type of matching, the Net-Net SBC does take the realm set in the local policy attributes into consideration. When the Net-Net SBC performs its regular matching method and you have enabled overlapping IP addresses for session agents, the Net-Net SBC might match session agents to different realms than the ones you intended when creating your configuration.

Local policy session agent matching provides a way to match session agents differently, taking realms and nested realms into consideration during the matching process. This difference is key to deployments with multiple peering partners that use the overlapping IP address feature, and have multiple local policies routing to the same IP address in different realms where some target next hops require session constraints but others do not. In the cases where no session constraints are required, session agents are not needed. But session agents still match the local policy, applying their constraints, because they match the next hop IP address.

In addition to modifying this behavior, this feature also affects the use of realms and nested realms. It triggers the use not only of realms, but of all the realms nested however deeply—thereby improving matching efficiency.

How It Works

You can set the local policy session agent matching option with values that define how the Net-Net SBC performs session agent matching:

- **any**—The Net-Net SBC looks up and stores matched session agents configured as next hops so it does not need to perform the lookup while processing requests, without regard to realms.

This behavior is the default when the SIP configuration does not have the local policy session agent matching option set.

- **realm**—The Net-Net SBC selects session agents in the realm that the local policy attribute indicates; this provides an exact match, rather than not taking the realm into consideration during session agent selection.

For example, the session agent is a match if the session agent **realm-id** and the local policy attribute **realm** parameters are an exact match.

- **sub-realm**—Session agents in the same realm or the same realm lineage—where session agents and realms are related to one another via realm parent-child relationships no matter the depth of realm nesting configured

For example, the session agent is a match if the local policy attribute **realm** is a sub-realm of the realm specified in the session agent **realm-id** parameter.

- **interface**—Session agents in the same realm or same realm lineage via the realm set in the local policy attribute, and whose realm uses the same signaling interface as the realm set in the local policy attribute

For example, the session agent is a match if the session agent **realm-id** is a sub-realm of the local policy attribute **realm**, and both referenced realms use the same SIP signaling interface.

- **network**—Session agents whose realm is in the realm lineage for the same realm set in the local policy attributes, and whose realm is associated with the same network interface as the realm set in the local policy attributes

For example, the session agent is a match if the session agent **realm-id** is a sub-realm of the local policy attribute **realm**, and realm reference by both use the same network interface.

If it cannot find a match, the Net-Net SBC will use the IP address as the next hop. Further, requests matching local policy attributes will not be associated with session agents, and so their constraints will not be applied.

The Net-Net SBC stores session agent information that it looks up when performing local policing session agent matching. To perform the lookup, it uses the session agent hostname as a key. When the hostname is an FQDN and there is a configured IP address in the **ip-address** parameter, the Net-Net SBC uses the ip-address value as a secondary key. Given this implementation, the following are true when selecting session agents:

- If multiple session agents share the same IP address, the one with an IP address in the hostname parameter takes precedence.
- If all session agents with the same IP address have an FQDN as their hostname, the one whose name is alphabetically lower will take precedence, where “alphabetically lower” means earlier in the alphabet (closer to A than to Z).
- For non-global session agents (whose realms are configured but not wildcarded) with an IP address, the Net-Net SBC uses a key that is a combination of the IP address and the realm in the form <address>:<realm>.
- For a session agent whose realm has a parent realm, the Net-Net SBC uses a combination of the IP address, realm, and realm-path (or lineage for the realm) in the form <address>:<realm-path>. For example, the realm path for a realm **core3** with a parent **core2**, which in turn has a parent core would be **core:core2:core3**.

When it looks up a session agent with a realm, the Net-Net SBC first searches for an exact match for the IP address and realm combination. If this fails, it performs a second search if the desired realm has parents or children. The Net-Net SBC locates an entry in its repository of session agent information that is greater than or equal to the IP address with the base realm, which is the ancestor of the desired realm without a parent. Having gathered this set of candidates, the Net-Net SBC narrows down the search for a match by comparing sub-realms and determines there is a match if either:

- The desired realm path is a sub-string of the entry’s realm path, or
- The entry’s realm path is a substring of the desired realm path (i.e., the desired realm is a sub-realm of the entry’s realm)

Then the Net-Net SBC orders the candidates by depth of the entry’s realm-path, or number of levels from the base realm relative to the depth of the desired realm. By searching the ordered set until the entry’s realm depth equals the desired realm’s depth, the Net-Net SBC determines a parent candidate, all subsequent entries being sub-realms of the desired realm. The Net-Net SBC only considers entries at the first

level deeper than the desired realm. If at this point there is only one entry, the Net-Net SBC deems it a match. Otherwise, it selects the parent candidate as the matching entry. In the event the search does not yield a matching realm, the Net-Net SBC uses the global session agent for the IP address, if there is one.

The following diagram shows the realm tree, where the clouds are realms and squares are session agents, representing a group of session agents sharing the IP address 1.2.3.4. The Net-Net SBC searches for the session agents lower in the tree along the session agent realm-path and the desired realm.



For the diagram above, the following shows how the hostname would look for this group of session agents.

Key	Session Agent (hostname[realm])
1.2.3.4 (This session agent owns the primary key for the IP address because its hostname is the IP address.)	1.2.3.4[CORE2]
1.2.3.4:CORE (IP+realm key entry)	SA[CORE]
1.2.3.4:CORE (IP+realm key entry)	1.2.3.4[CORE2]
1.2.3.4:CORE212 (IP+realm key entry)	SA212[CORE212]
1.2.3.4:CORE2121 (IP+realm key entry)	SA2121[CORE2121]
1.2.3.4:CORE231 (IP+realm key entry)	SA231[CORE231]
1.2.3.4:CORE232 (IP+realm key entry)	SA232[CORE232]
1.2.3.4:CORE: (IP+realm-path key entry)	SA[CORE]

Key	Session Agent (hostname[realm])
1.2.3.4:CORE:CORE2: (IP+realm-path key entry)	1.2.3.4[CORE2]
1.2.3.4:CORE2:CORE21:CORE212 (IP+realm-path key entry)	SA212[CORE212]
1.2.3.4:CORE2:CORE21:CORE212:CORE2121 (IP+realm-path key entry)	SA2121[CORE2121]
1.2.3.4:CORE2:CORE23:CORE231 (IP+realm-path key entry)	SA231[CORE231]
1.2.3.4:CORE2:CORE23:CORE232 (IP+realm-path key entry)	SA232[CORE232]

For each realm in the table above, the search results for each realm would look like this:

IP Address	Realm	Session Agent (hostname[realm])
1.2.3.4	CORE	SA[CORE]
1.2.3.4	CORE2	1.2.3.4[CORE2]
1.2.3.4	CORE21	SA212[CORE212]
1.2.3.4	CORE211	1.2.3.4[CORE2]
1.2.3.4	CORE212	SA212[CORE212]
1.2.3.4	CORE2121	SA2121[CORE2121]
1.2.3.4	CORE22	1.2.3.4[CORE2]
1.2.3.4	CORE23	1.2.3.4[CORE2]
1.2.3.4	CORE231	SA231[CORE231]
1.2.3.4	CORE232	SA232[CORE232]

ACLI Instructions and Examples

When you enable local policy session agent matching, remember that you can choose from five different ways to use the feature. The [How It Works \(320\)](#) section above explains your selections in detail, and they are: **all**, **realm**, **sub-realm**, **interface**, and **network**.

This example shows you how to use the **realm** selection.

To enable local policy session agent matching using the realm method:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-config** and press <Enter>.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. **options**—Set the options parameter by typing `options`, a <Space>, the option name `lp-sa-match=X` (where X is the local policy session agent matching method you want to use) with a “plus” sign in front of it. Then press <Enter>.

Remember that if you do not specify a method, the Net-Net SBC uses the **all** method.

```
ACMEPACKET(sip-config)# options +lp-sa-match=realm
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

SIP Per User Subscribe Dialog Limit

On a per-user basis, you can limit the number of SIP SUBSCRIBE dialogs by configuring the number allowed for a given realm. Without setting these limits, the Net-Net SBC permits an unlimited number of SUBSCRIBE dialogs. In some scenarios, endpoints might create multiple dialogs per event package, causing undesirable Net-Net SBC resource consumption.

You set the restrictions this feature uses in the enforcement profile configuration, which contains a subconfiguration called **subscribe-event**. Here, you enter the subscription event type and the limitations you want applied to it. You can enter more than one set of subscribe event limitations per enforcement profile. Once the enforcement profile contains the limitations you want, you then apply the enforcement profile to a realm and thereby restrict the number of SUBSCRIBE dialogs per user.

Limit Enforcement

Enabling this feature causes the Net-Net SBC, when it receives a 200 OK or NOTIFY for a corresponding SUBSCRIBE, to determine if the dialog created by the SUBSCRIBE would exceed the configured limit for a user for this event type. When it does, the Net-Net SBC removes the oldest dialog established for the user; usually, the oldest dialog has the lowest expiration timer. However, if a user (identified by the AoR) has more than AoC contact, then the Net-Net SBC will use the limit on a per-contact basis.

About Wildcarding

The Net-Net SBC supports wildcarding the event type in the **subscribe-event** configuration. To wildcard the value, you enter an asterisk (*) for the **event-type** parameter instead of typing in the name of an actual event type.

When you wildcard this value, the Net-Net SBC applies the subscription limitations you set across all event types. Or, if you have entered multiple subscribe-event configurations, the Net-Net SBC applies the wildcard limits across the event types for which you have not set limits.

Consider the following example of a configured enforcement profile with a wildcarded **subscribe-event** configuration:

```
enforcement-profile
  name rulefour
  allowed-methods
  sdp-address-check disabled
  subscribe-event
```

```

event-type *
max-subscriptions 1
subscribe-event
event-type xyz
max-subscriptions 0
last-modified-by admin@console
last-modified-date 2008-11-11 12:49:27

```

In this example, the enforcement profile allows all subscriptions that are event type xyz for a user. But it allows only one maximum for every other subscription event type.

ACLI Instructions and Examples

This section shows you how to configure an enforcement profile with a **subscribe-event** configuration. Remember that you can set up multiple **subscribe-event** configurations to correspond with the event types you want to control. It also shows you how to apply these limitations to a realm.

Setting Up Subscribe Dialog Limits

Setting up subscribe dialog limits means setting up an enforcement profile. For the sole purpose of setting up the subscription event limits, you only need to configure the name parameters and then as many **subscribe-event** configurations as you require. The enforcement profile has other uses, such as SIP SDP address correlation, so only configure the parameters you need.

To configure subscribe dialog limits:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **enforcement-profile** and press <Enter>.


```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement profile)#
```
4. **name**—Enter a name for this enforcement profile. You will use this name later when you apply the enforcement profile to a realm; it is the value you enter into the **enforcement-profile** parameter in the realm configuration.
5. Still in the enforcement profile configuration, type **subscribe-event** and press <Enter>.


```
ACMEPACKET(enforcement profile)# subscribe-event
ACMEPACKET(subscribe-event)#
```
6. **event-type**—Enter the SIP subscription event type for which you want to set up limits. You can also wildcard this value (meaning that this limit is applied to all event types except the others specifically configured in this enforcement profile). To use the wildcard, enter an asterisk (*) for the parameter value. See the [About Wildcarding \(324\)](#) section above for more information and a configuration example.

By default, this parameter is blank.

Note: The value you enter must be configured as an exact match of the event type expected in the SIP messages (except for the wildcard). Further, the value conforms to the event type BNF specified in RFC 3265.

7. **max-subscriptions**—Enter the maximum number of subscriptions allowed to a user for the SIP subscription event type you entered in the **event-type** parameter. Leaving this parameter set to 0 (default) means that there is no limit. You can set this parameter to a maximum value of 65535.

8. If you are entering multiple **subscribe-event** configurations, then you save them each by using the ACLI **done** command and then repeat Steps 6 and 7 to configure a new one. If you do not save each, then you will simply overwrite the first configuration repeatedly.

```
ACMEPACKET(subscribe-event)# done
```

9. When you finish setting up **subscribe-event** configurations and have saved them, exit to return to the enforcement profile configuration.

```
ACMEPACKET(subscribe-event)# exit
```

10. You also need to save the enforcement profile configuration.

```
ACMEPACKET(enforcement profile)# done
```

Applying an Enforcement Profile to a Realm

For the Net-Net SBC to use the limits you have set up, you need to apply them to a realm.

To apply an enforcement profile to a realm:

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

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)#
```

2. Type **media-router** and press <Enter>.

```
ACMEPACKET(configure)# media-router
```

```
ACMEPACKET(media-router)#
```

3. Type **realm-config** and press <Enter>. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.

```
ACMEPACKET(media-router)# realm-config
```

```
ACMEPACKET(realm-config)#
```

4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this realm. This value corresponds to the **name** parameter in the enforcement profile configuration. This parameter has no default value.

5. Save and activate your configuration.

STUN Server

The Net-Net SBC supports RFC 3489, which defines Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs). Known as STUN, this lightweight protocol allows applications to:

- Discover the presence and types of both NATs and firewalls between themselves and the public Internet
- Determine the public IP addresses allocated to them by the NAT

SIP endpoints use the STUN protocol to find out the public IP addresses and ports for SIP signaling and RTP media transport. Then they can use the address and port

information to create multimedia sessions with other endpoints on the public network.

You can define STUN servers functionality on a per-realm basis, allowing you set up multiple STUN servers.

About STUN Messaging

STUN messages uses six messages, three of which are used for Binding and three of which are uses for the Shared Secret. While it supports all three Binding messages (request, response, and error), the Net-Net SBC does not support the Shared Secret Request or the message integrity mechanism that relies on the shared secret. When acting as a STUN server, the Net-Net SBC responds to STUN binding requests in accordance with RFC 3489 and the rfc3489bis draft.

STUN messages can contain the following attributes:

Message Type	Attribute Description
MAPPED-ADDRESS	Appears in the Binding Response; contains the source IP address and port from which the Binding Request was sent to the STUN server.
XOR-MAPPED-ADDRESS	Appears in the Binding Response; contains the MAPPED-ADDRESS information encoded in a way the prevents intelligent NAT devices from modifying it as the response goes through the NAT.
SOURCE-ADDRESS	Appears in the Binding Response; contains the IP address and port from which the STUN server sent its response.
CHANGED-ADDRESS	Appears in the Binding Response; contains an alternate STUN server IP address and port, different from the primary STUN server port. The STUN client might use this attribute to perform the NAT tests described in RFC 3489.
CHANGE-REQUEST	Appears in the Binding Request; instructs the STUN server to send its response from a different IP address and/or port. The STUN client might use this attribute to perform the NAT tests described in RFC 3489.
RESPONSE-ADDRESS	Appears in the Binding Request; defines an IP address and port to which the STUN server should send its responses. Appears in the Binding Request;
REFLECTED-FROM	Appears in the Binding Response; reflects the IP address and port from which a Binding Request came. Only included when the Binding Request has used the RESPONSE-ADDRESS attribute.
UNKNOWN-ATTRIBUTES	Appears in the Binding Error; reflects the mandatory attributes in a Binding Request message that the server does not support.
ERROR-CODE	Appears in the Binding Error; indicates an error was detected in the Binding Request, and contains an error code and reason phrase.

To perform NAT discovery, the endpoint (STUN client) sends a Binding Request to the STUN server port (IP address and port) with which it is configured. The STUN server then returns either a;

- Binding Response—Allows the transaction to proceed
- Binding Error—Halts the transaction, and prompts the client to take the action appropriate to the response given in the ERROR-CODE attribute

When the transaction proceeds and the STUN server sends the Binding Response, that response contains the MAPPED-ADDRESS attribute, which contains the IP address and port from which the server received the request. The STUN client then uses the MAPPED-ADDRESS when sending signaling messages.

For example, a SIP endpoint sends Binding Requests from its SIP port to determine the public address it should place in SIP headers, like the Via and Contact, of the SIP requests it sends. When this SIP endpoint prepares to make or answer a call, it sends Binding Requests from its RTP port to find out the public address it should place in SDP included in an INVITE request or response.

STUN Server Functions on the Net-Net SBC

When the Net-Net SBC receives a STUN message, it first determines its message type. Only STUN Binding Requests are processed, and all other message types are dropped without response.

Then the Net-Net SBC examines the Binding Request's STUN attributes. It returns error responses if it finds any unsupported mandatory attributes. This takes the form of a Binding Error Response, containing the ERROR-CODE attribute with reason **420 (Unknown Attribute)** and an UNKNOWN-ATTRIBUTES attribute with a list of the unsupported attributes. If the Net-Net SBC receives a Binding Request with attributes that do not belong in STUN Binding Requests, it returns the Binding Error Response with the ERROR-CODE attribute with reason **400 (Bad Request)**.

Next the Net-Net SBC determines whether to follow RFC 3489 procedures or rfc3489bis procedures. If the Transaction ID contains the STUN cookie, then the Net-Net SBC follows rfc3489bis procedures; if not, it follows RFC 3489 procedures. Because it defines the procedures for testing the NAT to see what type of NAT it is, RFC 3489 procedures are most complex. Issues with reliability of those results have caused testing procedures and attributes to be deprecated in fc3689bis.

RFC 3489 Procedures

The Net-Net SBC (the STUN server) constructs the Binding Response and populates it with these attributes:

- MAPPED-ADDRESS and (optionally) XOR-MAPPED-ADDRESS—Containing the source IP address and port from which the server saw the request come
- SOURCE-ADDRESS—Containing the IP address and port from which the server will send the Binding Response
- CHANGED-ADDRESS—Containing the STUN server port that has a different address and different port from the ones on which the server request was received

If the Binding Request contains a RESPONSE-ADDRESS attribute, the server adds the REFLECTED-FROM attribute with the IP address and port from which the server saw the request come. Then the server sends the Binding Response to the IP address and port in the RESPONSE-ADDRESS attribute. If the RESPONSE-ADDRESS attribute's IP address and port are invalid, the server sends a Binding Error Response with an ERROR-CODE attribute reason **400 (Bad Request)** to the client.

If the Binding Request contains a CHANGE-REQUEST attribute, the server sends Binding Response from the IP address and port matching the information in the CHANGE-REQUEST. The following variations can occur:

- If the IP address and port flags are set, the server selects the server port with a different IP address and different port.
- If only the IP address flag is set, the server selects the server port with a different IP address but with the same port.
- If only the port flag is set, the server selects the server port with the same IP address but with a different port.

The selected server port appears in the Binding Responses's SOURCE-ADDRESS attribute. When there is no CHANGE-REQUEST attribute, the server uses the server port on which the Binding Request was received.

Finally, the server encodes the outgoing message and sends it to the client at either:

- The destination IP address and port in the RESPONSE-ADDRESS attribute, if it was present in the Binding Request.
- The MAPPED-ADDRESS.

rfc3489bis Procedures

If the Binding Request contains the appropriate cookie in its Transaction ID, the server constructs a Binding Response populated with the XOR-MAPPED-ADDRESS attribute. That attribute will contain the source IP address and port from which the server saw the request come. Then the server encodes and sends the message to the client from the IP address and port on which the request was received. The message is sent to the IP address and port from which the request came.

ACLI Instructions and Examples

You configure STUN servers on a per-realm basis, one server per realm. To support that various NAT tests it describes, RFC 3489 requires that two different IP addresses and two different UDP port numbers be used for each server. So your STUN server will listen on a total of four STUN server ports. Although newer work does away with this requirement, the Net-Net SBC supports it for the purpose of backwards compatibility.

For each realm configuration with an enabled STUN server, untrusted ACL entries will be added to forward all packets received on the four STUN Server Port.

To enable STUN server support for a realm:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

4. **stun-enable**—Set this parameter to **enabled** to turn STUN server support for this realm on. This parameter defaults to **disabled**, meaning STUN server support is off.
5. **stun-server-ip**—Enter the IP address for the primary STUN server port. The default for this parameter is 0.0.0.0. If you leave this parameter not configured, it defaults to the IP address of this realm's gateway.
6. **stun-server-port**—Enter the port to use with the **stun-server-ip** for primary STUN server port. The default is **3478**. This parameter is optional.
7. **stun-changed-ip**—Enter the IP address for the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. This IP address must be different from than the one defined for the **stun-server-ip** parameter. The default for this parameter is **0.0.0.0**. If you leave this parameter not configured, it defaults to the **stun-server-ip** value + 1.
8. **stun-changed-port**—Enter the port combination to define the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. The default for this parameter is **3479**. This parameter is optional.
9. Save and activate your configuration.

SIP Message Size Limiting

The Net-Net 9200 can constrain outgoing SIP messages to a maximum size in bytes in order to support fraud prevention techniques. If a message does exceed the configured size, it is dropped. A SIP message can be constrained from 0 to 65535 bytes, with a default value of 4096 bytes.

ACLI Configurations and Instructions

To set a maximum SIP message size to support fraud prevention:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. Type **select** to configure the existing sip config.


```
ACMEPACKET(sip-config)# select
```
5. Type **sip-message-len** <space> <length value> and press <Enter>. This completes the configuration.


```
ACMEPACKET(sip-config)# sip-message-len 5000
```
6. Save your work using the ACLI **done** command.

SIP SDP Address Correlation

SIP SDP address correlation ensures that when the Net-Net SBC receives a request containing SDP, the L3 source address of the request is compared against the address in the c-line of the SDP. When the addresses match, the session proceeds as it normally would. If there is a mismatch, the call is rejected with the default 488 status code. You can also configure the code you want to use instead of 488.

This functionality works only with non-HNT users. The value c=0.0.0.0 is an exception and is always processed.

ACLI Instructions and Examples

The **sdp-address-check**, in the **enforcement-profile** element can be set to enable the SDP address correlation.

To enable SDP address checking:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **enforcement-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# enforcement-profile
ACMEPACKET(enforcement-profile)#
```
4. Use the ACLI **select** command so that you can work with the enforcement profile configuration to which you want to add this parameter.


```
ACMEPACKET(enforcement-profile) select
```
5. **sdp-address-check**—Enable or disable SDP address checking on the Net-Net SBC. The default for this parameter is **disabled**.


```
ACMEPACKET(enforcement-profile)# sdp-address-check enabled
```
6. Save and activate your configuration.

If a mismatch occurs and you want to reject the call with a status code other than 488, you set the code you want to use in the SIP response code map entries which you map to a local response map.

To apply a new status code to a SDP address correlation mismatch:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-response-map** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-response-map
ACMEPACKET(response-map)#
```

4. Type name and a name for the local response map you are creating. Prepending the name with **local_** will help you organize which response maps are used as local response maps.

```
ACMEPACKET(response-map)# name local_response_map1
```

5. Type **entries** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(response-map)# entries
```

```
ACMEPACKET(response-map-entry)#
```

6. **recv-code**—Enter **sdp-address-mismatch** for which to apply the new status code.

7. **xmit-code**—Enter the new status code you want to use.

8. **reason**—Enter the reason to correlate with the new status code.

```
ACMEPACKET(response-map-entry)# recv-code sdp-address-mismatch
```

```
ACMEPACKET(response-map-entry)# xmit-code 403
```

```
ACMEPACKET(response-map-entry)# reason sdp address mismatch
```

9. Save and activate your configuration.

You apply this feature per-realm by setting the enforcement profile parameter for a realm.

To apply an enforcement profile to a realm:

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

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)#
```

2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(configure)# media-manager
```

```
ACMEPACKET(media-manager)#
```

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

```
ACMEPACKET(media-manager)# realm-config
```

```
ACMEPACKET(realm-config)#
```

4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this realm.

```
ACMEPACKET(realm-config)# enforcement-profile profile1
```

5. Save and activate your configuration.

SIP Unauthorized Endpoint Call Routing

The Net-Net SBC can route new dialog-creating SIP INVITEs from unauthorized endpoints to a session agent or session agent group; then rejection can occur based on the allow-anonymous setting for the SIP port. This type of provisional acceptance and subsequent rejection applies only to INVITEs; the Net-Net SBC continues to reject all other requests, such as SUBSCRIBE.

You might enable this feature if you have a network in which unauthorized SIP endpoints continually try to register even if the Net-Net SBC has previously rejected them and never will accept them. For instance, the user account associated with the endpoint might have been removed or core registrars might be overloaded.

ACLI Instructions and Examples

You enable the routing of unauthorized endpoints to session agents and session agent groups that will reject them in the SIP interface configuration.

To enable SIP unauthorized endpoint call routing:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **route-unauthorized-calls**—Enter the name (or IP address) of the session agent or session agent group to which you want calls from unauthorized endpoints routed. This parameter is blank by default, meaning the SIP unauthorized call routing feature is disabled.

Remember your settings in the **allow-anonymous** parameter in the SIP port configuration provide the basis for rejection.

5. Save and activate your configuration.

SIP Registration Failure Mapping

You can configure the Net-Net SBC to remap a SIP failure response—which it receives from another network device or that it generates locally—to a 200 OK. You might want the Net-Net SBC to perform this type of mapping for circumstances where non-malicious endpoints continually attempt registration, but will stop (and still not be registered) when they receive a 200 OK. This response mapping does not actually register the client with the Net-Net SBC, meaning that there is neither a registration cache entry or a CAM ACL for it.

For the 200 OK it generates, the Net-Net SBC removes any Reason or Retry-After header in the 200 OK and sets the expires time. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). You can also set this value using the register-response-expires parameter, but the value you set should never exceed the Register request's expires time.

ACLI Instructions and Examples**To enable SIP registration failure code mapping:**

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# sip-response-map
```

```
ACMEPACKET(response-map)#
```

4. Type **entries** and press <enter>

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

5. **method**—Enter the name of the received SIP failure response message you want to map to a 200 OK. Currently, this feature supports the **REGISTER** method.
6. **register-response-expires**—Enter the time you want to use for the expires time what mapping the SIP method you identified in the method parameter from Step 4. The maximum is 999999. By default, the expires time is the Retry-After time (if there is one in the response) or the expires value in the Register request (if there is no Retry-After expires time). Any value you configure in this parameter (when not using the defaults) should never exceed the Register request's expires time.
7. Save and activate your configuration.

SIP SDP Anonymization

In order to provide an added measure of security, the Net-Net SBC's topology-hiding capabilities include SDP anonymization. Enabling this feature gives the Net-Net SBC the ability to change or modify certain values in the SDP so that malicious parties will be unable to learn information about your network topology.

To do this, the Net-Net SBC hides the product-specific information that can appear in SDP **o=** lines and **s=** lines. This information can include usernames, session names, and version fields. To resolve this issues, the Net-Net SBC makes the following changes when you enable SDP anonymization:

- Sets the session name (or the **s=** line in the SDP) to **s=-**
- Sets the username in the origin field to **-Net-Net SBC**
- Sets the session ID in the origin field to an integer of incrementing value

Note that for mid-call media changes, the session identifier is not incremented.

To enable this feature, you set a parameter in the media manager configuration.

ACLI Instructions and Examples

To enable SDP anonymization:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-manager** again to access the media manager configuration, and press <Enter>.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **anonymous-sdp**—Set this parameter to **enabled** to use the SDP anonymization feature. When you leave this parameter empty the feature is turned off. The default value is **disabled**.
5. Save and activate your configuration.

SIP Accounting

This section explains SIP accounting using the Remote Authentication Dial In User Service (RADIUS) Accounting System (RAS).

For accounting purposes, the Net-Net system uses RADIUS to send accounting messages. These messages are transmitted to one of a predefined list of accounting servers using a predefined forwarding strategy. RAS provides a mechanism for temporarily storing session initiation and completion statistics and for delivering these statistics to accounting servers located elsewhere in the network.

See the *Net-Net 9200 Radius Guide* for details.

SIP Session Accounting

The RAS client can record SIP session activity based on configuration and a CDR. The CDR determines which messages are generated and determines the RADIUS attributes included in the messages. The RAS client must be capable of sending CDRs to any number of RADIUS accounting servers, using the defined hunt, failover, round robin, fewest pending, or fastest server strategies.

The establishment, failed establishment, change, or removal of a SIP session can trigger RADIUS Accounting Request messages. The RAS might also send notification of its status (enabled/disabled). RADIUS Accounting Request messages include the following:

- **Start** – SIP session has started.
- **Interim-Update** – SIP session parameters have changed.
- **Stop** – SIP session has ended.
- **Accounting-On** – Creation of a new RADIUS client.
- **Accounting-Off** – RADIUS client has shut down.

Each SIP session might generate Start, Interim-Update, and Stop messages based on the local configuration when the session is initiated. Each Start message tells the RADIUS server that a SIP session has started. Each Interim-Update message changes the SIP session parameters, and may report the session characteristics for the SIP session to that point. Each Stop message informs the RADIUS server that a SIP session has ended and reports session characteristics.

The RAS has the ability to transmit all RADIUS messages related to a SIP session at the end of the session, regardless of which messages are generated and when they are generated. Some customers might choose this option to reduce the likelihood of the RADIUS messages being logged to different servers, or in different log files on the same server.

The RAS always generates a RADIUS Stop message when the SIP session ends, regardless of the SIP session termination cause. The termination cause and the session characteristics are reported.

RADIUS Messages

The following table identifies the relationship between the SIP elements and the RADIUS attributes included in Accounting Request messages to the RADIUS server.

RADIUS Attribute	SIP Data Element	Message
NAS IP-Address	SIP proxy IP address Either (or both) the NAS IP address/port or the NAS identifier should be included. The values reflect the NAS (RADIUS client). The source IP address of the packet (not the fields within the packet) is used to select the secret and address the response message.	Start, Interim-Update, Stop, On, Off
NAS Port	SIP proxy port	Start, Interim-Update, Stop, On, Off
NAS Identifier	SIP proxy Either the NAS IP address/port or the NAS Identifier should be included (both may be included)—the values reflect the NAS (RADIUS client). The source IP address of the packet (not the fields within the packet) is used to select the secret and address the response message.	Start, Interim-Update, Stop, On, Off
User-Name	Optional	Start, Interim-Update, Stop, On, Off
Service Type	Type of service.	Start, Interim-Update, Stop, On, Off
Acct-Session-ID	Call-ID field of the SIP INVITE, or RADIUS client information	Start, Interim-Update, Stop, On, Off
Acct-Multi-Session-ID	Includes an additional identifier that ties together multiple SIP sessions (for example, a conference call or bundled flows).	Start, Interim-Update, Stop, On, Off
Called Station ID	To field of the SIP INVITE	Start, Interim-Update, Stop
Calling Station ID	From field of the SIP INVITE	Start, Interim-Update, Stop
Acct-Terminate-Cause	Reason for session ending (refer to section Session Termination)	Stop, Off
Acct-Session-Time	Length of session	Interim-Update, Stop, Off
Acct-In-Packets	Number of packets transferred	Interim-Update, Stop, Off
Acct-Output-Packets	Number of packets transferred	Interim-Update, Stop, Off
Acct-Input-Octets	Number of bytes transferred	Interim-Update, Stop, Off
Acct-Output-Octets	Number of bytes transferred	Interim-Update, Stop, Off

Session Termination

Sessions are terminated for reasons that include normal termination, signaling failure, timeout, or network problems. The following table maps RADIUS accounting termination cause codes to network events.

RADIUS Termination Cause	Event	Message
User request	SIP BYE message	Stop
User error	SIP signaling failed to establish session (accompanied by disconnect cause)	Stop
NAS request	RADIUS server disabled	Off

**SIP CDR
Generating
Behaviors**

The following accounting behaviors are all enabled at once by setting the **enhanced cdr** option in the SIP config:

- The Net-Net SBC generates a CDR stop record when a call is rejected due to exceeding session constraints. For example: calls rejected with "500 Outbound Constraints Exceeded" generate stop CDRs.
- The Net-Net SBC generates a CDR stop record when a call is rejected due to not finding a route. For example: calls rejected with "404 No Route Found" generate stop CDRs.
- The Net-Net SBC generates a CDR record for all cases where it rejects a well-formed request.

ACLI Instructions

To enable CDR handling of the above SIP scenario:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. Set the options parameter by typing **options**, a <Space>, the option name **strip-route-headers** with a "plus" sign in front of it, and then press <Enter>.


```
ACMEPACKET(sip-config)# options +enhanced-cdr
```

If you type **options enhanced-cdr** (without the "plus" sign), you will overwrite any previously configured options. In order to append the new option to the list, you must prepend the new option with a "plus" sign as shown in the previous example.
5. Save and activate your configuration.

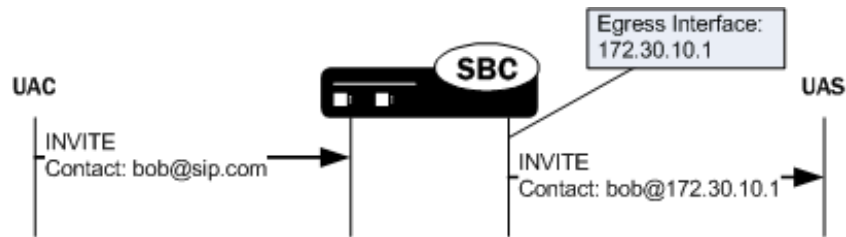
SIP GRUU

SIP Globally Routable User Agent (UA) URIs (GRUU) are designed to reliably route a SIP message to a specific device or end user. This contrasts with a SIP AoR which can refer to multiple UAs for a single user, thus contributing to routing confusion. The Net-Net SBC can perform different behaviors when it finds SIP GRUUs in Contact headers.

User agents supporting GRUU include a GRUU-identifying parameter in the Contact header of a dialog forming and target refresh requests. The Net-Net SBC scans for the GRUU parameter in the Contact header only when the endpoint it receives a request from is registered or when the pass-gruu-contact parameter is enabled.

Contact Header URI Replacement

When no GRUU is encountered in the contact header, and when a SIP message is forwarded to the egress realm, the hostname, as received in the contact header's URI, is replaced with the Net-Net SBC's egress interface. For example:



When the Net-Net SBC forwards a request where the original Contact header contains a GRUU, the contact header's URI is forwarded unchanged on the egress side of the call. For example:



Record-Route Addition

When the request is forwarded to a realm where the endpoint's registrar does not exist, the Net-Net SBC adds a Record-Route header containing the egress SIP interface address. This causes subsequent replies or requests addressed to the GRUU to be routed through the SBC first.



When the request is forwarded to the realm where the registrar exists, adding the Record-Route header is unnecessary and does not occur. This is because subsequent requests are directed to the registrar which will ultimately forward them to the Net-Net SBC using the registered Contact in the Request-URI.

GRUU URI Parameter Name

The Net-Net SBC scans for a **gr** URI parameter in the contact header to identify it as a GRUU as defined in the ietf draft[2]. The Net-Net SBC can be configured to scan for a **gruu** URI parameter in the contact header too. This alternate behavior is enabled with the scan-for-ms-gruu option and is used to interact with the Microsoft Office Communications Server unified communications product. When "scan-for-ms-gruu" is enabled, the Net-Net SBC scans first for the **gruu** URI parameter. If not found, it then scans for **gr** URI parameter.

ACLI Instructions and Examples

This section shows you how to configure the GRUU support for non-registered contacts. Enabling GRUU functionality to parse for **gr** URI parameter rather than the IETF standard **gruu** parameter is also provided.

To configure SIP GRUU functionality:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI **select** command) before making your changes.
4. **pass-gruu-contact**—Set this parameter to enabled to parse for **gr** URI parameter in the contact header in non-registered endpoints' messages and then pass the messages through the system.
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **scan-for-ms-gruu**. This option forces the Net-Net SBC to first scan for the **gruu** URI parameter, then the **gr** URI parameter.
6. Save and activate your configuration.

SIP Proxy Mode High Subscription-Dialog Capacity

Rich Communications Service (RCS) is an initiative that aims to provide IM, presence, file sharing, and video-share capability based upon mobile phone number addressing. Applications such as IM require a larger number of SUBSCRIBE dialogs per user (each buddy list member could require 1 SUBSCRIBE dialog, thus it would be reasonable to estimate each user requiring 100 SUBSCRIBE dialogs). With the Net-Net SBC's default limit of 250,000 dialog capacity, RCS requires an increase in this number.

To support RCS, the Net-Net SBC's SIP Proxy mode high subscription capacity dialog functionality increases subscription dialogs to more than 5 million. This is accomplished by flagging subscription dialogs to be proxied in transaction stateful proxy mode, a less resource intensive proxy mode. In addition, the Net-Net SBC maintains topology hiding for Via, Contact, and Record-Route headers related to the subscription dialogs. Also the Net-Net SBC obfuscates any URIs contained in application/pdf+xml message bodies.

Subscription Requests

For all Event Packages, as indicated in the Event header of a PUBLISH, SUBSCRIBE or NOTIFY request that the Net-Net SBC should proxy and not maintain state, configure them as values in the **proxy sub events** parameter in the sip config. This configuration is applicable to REFER request too, although the Net-Net SBC looks to the refer header. This means that PUBLISH, SUBSCRIBE, NOTIFY, and REFER messages, with event/refer types matching those configured are processed in transaction-stateful mode requests, while other requests are processed in B2BUA mode.

Note: The PUBLISH request is not a dialog creating request, but topology hiding needs to be applied to the PIDF (application/pidf+xml Content-Type) so that subsequent NOTIFY requests containing portions of the published PIDF can be properly decoded.

The per-user SUBSCRIBE dialog limit feature is not compatible with subscriptions handled in proxy mode.

Topology Hiding

The Net-Net SBC encodes the original, received URI "hostport", and places it into the URI's gr= parameter (GRUU) on egress from the system. Similar URI encoding is performed on the message body for Content-Type "application/pidf+xml". If a URI gets encoded multiple times, the additional encodings are delimited by a plus sign.

ACLI Instructions

To configure high subscription dialog capacity in the SIP config:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
```

Type select to select the sip config for editing.
4. **proxy-sub-events**—Enter the event package types (or refer header contents) that the Net-Net SBC should proxy in transaction mode only. Use the add and delete subcommands to add or delete one of multiply configured events types. For example:


```
ACMEPACKET(sip-config)# proxy-sub-events add presence
```
5. Save and activate your configuration.

SIP Diversion to SIP-ISUP Interworking

For networks in which there are devices that do not support SIP-T or SIP-I (and support native SIP alone), the Net-SBC now supports SIP Diversion interworking. This feature enables such devices to function properly in instances that require SIP-T/SIP-I style ISUP IAM message encapsulation in ISUP requests, and to receive any call forwarding information in the IAM according to ISUP standards.

The Net-Net SBC interworks a native SIP INVITE request to SIP-T one by inserting an ISUP IAM body based on the INVITE; this includes redirections information based on the Diversion header. This feature can also perform the reverse translation. That is, it can interwork a SIP INVITE that does have the ISUP IAM body to a non-ISUP INVITE. In this case, the Net-Net SBC generates the necessary Diversion headers based on the IAM's Redirection information.

ACLI Instructions and Examples

To use this feature, you set up:

- **sip-profile**—Defines the redirection behavior

- **sip-isup-profile**—Defines the ISUP version to use when supporting SIP-T

You can then apply these profiles to realms, session agents, and SIP interfaces where you want this feature enabled.

Configuring a SIP Profile

The **sip-profile** configuration contains information that defines redirection behavior for the configuration where you apply it. You can set the redirection behavior to: **none**, **isup**, **inherit**, or **redirection**. You also uniquely identify the profile so that you can apply it by name in other configurations. This is a multiple-instance configuration, meaning that you can set up as many SIP profiles as needed.

To set up a SIP profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-profile** and press <Enter>.


```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```
4. **name**—Enter the name of the SIP profile. You will use this name when you apply this profile to realms, session agents, and SIP interfaces. This parameter is required, and it has no default value.
5. **redirection**—Choose the redirection mode you want to use: **none** (default), **isup**, **inherit**, or **redirection**. Note that when you set this parameter to **isup**, you should configure along with it a SIP ISUP profile; this will avoid any possible incompatibility when support for this feature expands (as expected).
6. Save your work.

Configuring a SIP ISUP Profile

The **sip-isup-profile** configuration contains information about what ISUP version you want to use. You also uniquely identify the profile so that you can apply it by name in other configurations. This is a multiple-instance configuration, meaning that you can set up as many SIP ISUP profiles as needed.

To set up a SIP ISUP profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-isup-profile** and press <Enter>.


```
ACMEPACKET(session-router)# sip-isup-profile
ACMEPACKET(sip-isup-profile)#
```
4. **name**—Enter the name of the SIP ISUP profile. You will use this name when you apply this profile to realms, session agents, and SIP interfaces. This parameter is required, and it has no default value.

5. **isup-version**—Specify the ISUP version you want used in this profile in order to support SIP-T: **ansi-2000** (default) or **itu-99**.
6. Save your work.

Applying SIP Profiles and SIP ISUP Profiles

When you want to enable this feature for a realm, session agent, or SIP interface, you configure the **sip-profile** and **sip-isup-profile** parameters with the name of the profile you want applied.

The sample here shows this feature being applied to a session agent, but the realm and SIP interface configurations also have the same two parameters you use to set up the feature.

To apply a SIP profile and a SIP ISUP profile to a session agent:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **sip-profile**—Enter the name of the SIP profile, which defines the redirection behavior. This is the value you entered in the **name** parameter of the SIP profile configuration. This parameter has no default value.
5. **sip-isup-profile**—Enter the name of the SIP ISUP profile, which defines the ISUP version to use for SIP Diversion SIP-ISUP interworking. This is the value you entered in the **name** parameter of the SIP ISUP profile configuration. This parameter has no default value.
6. Save your work.

SIP-ISUP Format Version Interworking

ISUP message can be carried in SIP messages through either a standard body or through a multipart MIME encoded body. While ANSI and ITU are the two major groups, each contains many specific variants. To facilitate instances where two sides of a call use different versions, the Net-Net SBC supports interworking between the following SIP ISUP formats: ANSI, ITU, ETSI-356 (an ITU variant), and GR-317 (an ANSI variant). To do so, the Net-Net SBC can move, delete, and add parameters to various sections of the message.

Details

The ISUP message version is determined by one of two things: the content type of the SIP message or the MIME content-type. When the base and version parameters do not match, the Net-Net SBC first uses the base parameter value to determine the format. If there is no base, the Net-Net SBC then checks the version parameter. And if there is neither, the Net-Net SBC uses the **isup-version** configured in the **sip-isup-profile** configuration from the inbound realm, session agent, or SIP interface. Available values for that parameter are **ansi-2000**, **itu-99**, **gr-317**, or **etsi-356**. The Net-Net SBC considers unknown any value for the version that fails to match one of these or is missing.

Messages that contain an unknown ISUP format pass through the Net-Net SBC untouched. If there are operations to be performed on them, however, SIP ISUP HMR will take place. After the body has been converted, the Net-Net SBC updates both the base and version parameters of the content-type.

Custom formats are not supported.

ACLI Instructions and Examples

This section shows you how to set up a SIP-ISUP format interworking. First, you configure a SIP ISUP profile, and then you apply it to a realm, session agent or SIP interface.

Configuring a SIP ISUP Profile

To set up a SIP ISUP profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-isup-profile** and press <Enter>.


```
ACMEPACKET(session-router)# sip-isup-profile
ACMEPACKET(sip-isup-profile)#
```
4. **name**—Enter the name of the SIP ISUP profile. You will use this name when you apply this profile to realms, session agents, and SIP interfaces. This parameter is required, and it has no default value.
5. **isup-version**—Specify the ISUP version to which you want to convert: **ansi-2000**, **itu-99**, **gr-317**, or **etsi-356**.
6. **convert-isup-format**—Set this parameter to **enabled** if you want to perform SIP ISUP format version interworking. The default is **disabled**, meaning that this feature is turned off.
7. Save your work.

Applying SIP ISUP Profiles

When you want to enable this feature for a realm, session agent, or SIP interface, you configure the **sip-isup-profile** parameter with the name of the profile you want applied.

The sample here shows this feature being applied to a session agent, but the realm and SIP interface configurations also have the same parameter you use to set up the feature.

To apply a SIP profile and a SIP ISUP profile to a session agent:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.

```
ACMEPACKET(session-router)# session-agent
```

```
ACMEPACKET(session-agent)#
```

4. **sip-isup-profile**—Enter the name of the SIP ISUP profile, which defines the ISUP version to convert to. This is the value you entered in the **name** parameter of the SIP ISUP profile configuration. This parameter has no default value.
5. Save your work.

HMR for SIP-ISUP

The Net-Net SBC's HMR functionality can operate on ISDN user party (ISUP) binary bodies. Using the same logic and mechanisms that are applied to SIP header elements, HMR for SIP-ISUP manipulates ISUP parameter fields and ISUP message parts. You can create MIME rules that function in much the same way the SIP header rules do; whereas SIP header rules can change the specific headers of a SIP message, MIME rules can manipulate targeted body parts of a SIP message.

Changes and Additions to Equality Operators

The following table defines the additions and changes to HMR equality operators introduced with this feature.

Unlike the Boolean operators the ampersand (&) and the pipe (|), you can use the following equality operators in conjunction with string operators. For example, a header-value with its **comparison-type** set to **boolean**, can have this match-value evaluated: "(\$rule1.\$elem1.\$0 + \$rule1.\$elem2.@1) == \$rule2.\$0". Equality operators can also be used with Boolean operators, as in this example: "(\$rule1.\$0 == \$rule2.\$1) & \$rule3".

Equality operators always evaluate to either true or false.

Equality Operator Symbol	Short Description	Detailed Information
==	String case sensitive quality operator	Performs a character-by-character, case-sensitive string comparison on both the left side and the right side of the operator.
~=	String case insensitive quality operator	Performs a character-by-character, case-insensitive string comparison on both the left side and the right side of the operator.
!=	String case sensitive not equal to equality operator	Performs a character-by-character, case-sensitive string comparison on both the left side and the right side of the operator, returning true if the left side is equal to or less than the right side of the operator.
>=	Greater than or equal to operator	Performs a string-to-integer conversion. If the string-to-integer comparison fails, the value is treated as 0. After the conversion, the operator will compare the two values and return true only if the left side is greater than or equal to the right side of the operator.

Equality Operator Symbol	Short Description	Detailed Information
<	Less than operator	Performs a string-to-integer conversion. If the string-to-integer conversion fails, the value is treated as 0. After the conversion, the operator will compare the two values and return true only if the left side is less than the right side of the operator.
>	Greater than operator	Performs a string-to-integer conversion. If the string-to-integer conversion fails, the value is treated as 0. After the conversion, the operator will compare the two values and return true only if the left side is greater than the right side of the operator.

Reserved Words

To improve system performance and simplify configuration, the Net-Net SBC now supports pre-defined reserved words for commonly-used URI parameters for HMR.

Reserved words retrieve values directly from the SIP message, without your needing to create rules to store them. Their function is similar to the `$REMOTE_VIA_HOST` and other already-defined variables. If the header or value does not exist in the SIP message, either an empty string is returned or—for Boolean uses—the value `FALSE` is returned.

Reserved words apply to these commonly-accessed SIP headers and their prefixes are:

- To—`$TO_xxx`
- From—`$FROM_xxx`
- Contact—`$CONTACT_xxx`
- Request URI—`$RURI_xxx`
- P-Asserted-Identity—`$PAI_xxx`
- P-Preferred-Identity—`$PPI_xxx`
- P-Called-Party-ID—`$PCPID_xxx`

The following table contains the list of supported reserved words and a description of each.

Reserved Word	Description
<code>xxx_USER</code>	The URI name of the header without any user parameters
<code>xxx_PHONE</code>	The URI user of the header as a phone number but without visual separators; may or may not contain a leading plus sign (+)
<code>xxx_HOST</code>	The URI host of the header
<code>xxx_PORT</code>	The URI port of the header; Value set to 5060 even if it is not actually in the message
<code>CALL_ID</code>	Resolves to the Call-ID of the current SIP message; added for convenience, and is a common store rule
<code>TIMESTAMP_UTC</code>	Timestamp is RFC 3339 format: 2009-10-10T22:00:09Z or YYYY-MM-DDTHH:MM:SS.PPPZ. The .PPP refers to partial seconds and is optional; time is based on UTC.

The reserved word CRLF resolves to “\r\n” and is commonly used in MIME manipulation. If you are creating a new body, there might be a need for many CRLFs in the new-value parameter.

All of these operators cause additional overhead to the HMR processing because each operator requires an evaluation of the left and right sides of the expression. To speed up evaluation of new-value expressions, you can now enter escapable characters (\f, \n, \r, \t, \v) with a backslash (\) and the Net-Net SBC will convert them to escaped characters during the compilation of the expression (i.e., ACLI configuration time).

Changes to Action

In releases prior to S-D7.1.0, the **sip-manip** action is only supported in the header rule (**header-rule**) configuration. This limitation has been deemed unnecessary, and so you can now set the action parameter to **sip-manip** at all levels of HMR configuration, including element rules.

About MIME Rules

MIME rules (set up in the ACLI **mime-rules** configuration) operate much the same way that SIP header rules do. You can set parameters in the MIME rules that the Net-Net SBC uses to match against specific SIP methods and message types. The system compares the search criteria against the body or body parts using the type of comparison you choose. Offering a variety of selection, you can pick kind of manipulation that suits your needs; the Net-Net SBC then takes action with matching and new values to change the SIP message.

You identify the MIME rule by configuring a content type that refers to the specific body part on which to operate. For example, given a SIP Content-Type header with the value `multipart/mixed;boundary=unique-boundary-1`, you would enter a content-type value of **application/sdp** to manipulate specifically on the SDP portion of the SIP message. The Net-Net SBC knows automatically if it is operating on SIP messages with single or multiple body parts, and the content-type setting applies to both kinds. And when making its comparison, the Net-Net SBC matches the content-type of the body with regard to case (case insensitive), ignoring any header parameters.

Both for making comparisons against the body part and for new/replacement values, the Net-Net SBC treats the match and new values you set for a MIME rule as ASCII strings. Therefore, a mime rule operating on a binary body part will yield an improper conversion of a new value with respect to the binary body part.

Within MIME rules, you configure MIME headers, which operate on the specific headers in the match body part of the SIP message. The Net-Net SBC uses the MIME header name to run a string comparison to match the specific header in the message's body part.

Using these rules, you can also manipulate the preamble—or the SIP message text that follows the headers but precedes the body separator. To do so, enter the keyword **@preamble** for the content type parameter in the MIME rule. Likewise you can manipulate the epilogue—or the text that follows the last body part after the last separator—using the keyword **@epilogue**.

Note that the ACLI limits character entries to 255 characters before the return character must be entered, but MIME parts can easily exceed this 255-character size. So you might need to enter a value larger than 255 characters. To do so, you start your entry (in the match-value or new-value parameters) with a plus sign (+). The plus

sign instructs the system to add the string after it to the pre-existing match or new value. For the new-value parameter, the Net-Net SBC checks the value immediately for validity. Be sure that when you are appending values to a new-value that the entire expression is valid at each point where strings are appended.

ACLI Instructions and Examples

This section shows you how to configure MIME rules and MIME headers.

To configure MIME rules:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **mime-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# mime-rules
ACMEPACKET(sip-mime-rules)#
```
5. **name**—Enter a name for this MIME rule. This parameter is required and has no default.
6. **content-type**—Enter the content type for this MIME rule. This value refers to the specific body part in the SIP message body that is to be manipulated. For example, given a SIP Content-Type header with the value `multipart/mixed;boundary=unique-boundary-1`, you would enter a content-type value of **application/sdp** to manipulate specifically on the SDP portion of the SIP message.

To manipulate the SIP preamble or epilogue, enter the keyword **@preamble** or keyword **@epilogue**.
7. **action**—Choose the type of action you want to be performed: **none**, **add**, **delete**, **manipulate**, **store**, **sip-manip**, and **find-replace-all**. These are the same actions you can select when configuring SIP header manipulation. The default is **none**.
8. **comparison-type**—Enter the way that you want body part of the SIP message to be compared. This choice dictates how the Net-Net SBC processes the match rules against the SIP header. the default is **case-sensitive**. The valid values are: **case-sensitive**, **case-insensitive**, **boolean**, **refer-case-sensitive**, **refer-case-insensitive**, and **pattern-rule**.
9. **msg-type**—Enter the SIP message type on which you want the MIME rules to be performed. Valid values are **any**, **request**, and **reply**. The default value is **any**.
10. **methods**—Enter the list of SIP methods to which the MIME rules applies. There is no default for this parameter.
11. **match-value**—Enter the value to match against the body part in the SIP message. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators.

12. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute.

To configure MIME headers for performing HMR operations on specific headers in the matched body part of the SIP message:

1. Follows Steps 1 through 4 above.
2. Type **mime-header-rules** and press <Enter>.


```
ACMEPACKET(sip-mime-rules)# mime-header-rules
ACMEPACKET(sip-mime-header-rules)#
```
3. **name**—Enter a name for this MIME header rule. This parameter is required and has no default.
4. **mime-header**—Enter the value to be used for comparison with the specific header in the body part of the SIP message. There is no default for this parameter.
5. **action**—Choose the type of action you want to be performed: **none**, **add**, **delete**, **manipulate**, **store**, **sip-manip**, and **find-replace-all**. The default is **none**.
6. **comparison-type**—Enter the way that you want the header in the body part of the SIP message to be compared. This choice dictates how the Net-Net SBC processes the match rules against the SIP header. the default is **case-sensitive**. The valid values are: **case-sensitive**, **case-insensitive**, **boolean**, **refer-case-sensitive**, **refer-case-insensitive**, and **pattern-rule**.
7. **match-value**—Enter the value to match against the header in the body part of the SIP message. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators.
8. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute.
9. Save your work.

About MIME ISUP Manipulation

MIME ISUP manipulation supports performing HMR operations on SIP ISUP binary bodies, and is configured in the **mime-isup-rule** configuration. This configuration works the same way that the MIME rule configuration does and contains the same parameters for you to set, but it also includes additional parameters and a sub-configuration targeted specifically for ISUP application.

Net-Net SBC MIME ISUP Parameters

- **isup-msg-type**—Refers to specific ISUP message types (such as IAM and ACM). The Net-Net SBC uses with the msg-type parameter (which identifies the SIP message) in the matching process. You enter values in this parameters as a list of numbers rather than as an enumerated value because of the large number of ISUP message type, and the range is between 0 and 255.
- **isup-spec**—Specifies how the Net-Net SBC is to parse the binary body; valid values are the enumerated type. The values for this parameter are these SIP ISUP formats
 - ANSI-2000—Corresponding to ANSI T1.113-2000
 - ITU-99—Corresponding to ITU Q.763

Because ISUP messages do not identify their format, you must designate which you want to use.

- **isup-parameter-rules** (sub-configuration)—If you are familiar with HMR, then think of this parameter as being similar to the element-rule for a SIP header rule. You use it to create, manipulate, and store different parameters in the body of the ISUP message. Two parameters for this rule are unique: **parameter-rule** and **parameter-format**.
- **parameter-rule**—Using ISUP parameter mapping, this setting identifies which of the ISUP parameters on which you want to perform manipulation. This parameter takes values between 0 and 255, and you must know the correct ISUP mapping value for your entry. The Net-Net SBC calculates the offset and location of this parameter in the body. Note that the value returned from the body does not the type or length, only the parameter value. For example, a parameter-type value of 4 acts on the Called Party Number parameter value.

In accordance with the ISUP specifications, only certain message types are allowed to have optional parameters. And if optional parameters are present, an offset field must exist for them; so its value is 0 even if there are no optional parameters in the SIP message. For example, if you define a SIP ISUP rule that applies to all message types and adds a parameter that is neither fixed nor variable, The Net-Net SBC adds it as an optional parameter regardless of whether that message type should not support optional parameters.

If you define an ISUP parameter rule with an **add** action and an empty **new-value**, the Net-Net SBC uses the default for that parameter. If you define an ISUP parameter rule with a **replace** action and no parameters exist, the Net-Net SBC will not perform any action. This behavior is consistent with that of SIP header rules in that a value can only be replaced if it already exists. If there is a value and no new value is set, the Net-Net SBC set it as a zero-length parameter.

- **parameter-format**—This parameter converts the specific parameter to a string representation of that value. Valid values for **parameter-format** are: **number-param**, **hex-ascii** (default), **binary-ascii**, **ascii-string**, and **bcd**.

Both match and new values are encoded and decoded by the designated parameter-format type. In this regard, the **match-value** decodes the parameters and the **new-value** encodes the ASCII string into the respective binary format.

Note if you enter a new-value setting larger than the size of the parameter, the Net-Net SBC will perform no operation and will generate a corresponding error log message.

The following table provides information about the values you can enter:

parameter-format Setting	Description
hex-ascii	Default. Converts the entire binary body. Non-hexadecimal characters fail in matching against the body part if they are in the match-value setting and non-hexadecimal characters places in a new-value setting result in no operation being formed.
binary-ascii	Converts each hexadecimal value to its corresponding string, binary representation. For example, the Net-Net SBC would convert the ISUP parameters with a hexadecimal binary value of 8A to 10001010. Non-binary digit characters fail when matching against the body part if they are contained in the match-value setting and non-binary characters in the new-value setting results in no operation being performed.
ascii-string	Treats the binary parameter as true ASCII in raw format. The Net-Net SBC supports only the printable range of ASCII characters. If a value in the ISUP parameters cannot be decoded to ASCII, the Net-Net SBC returns an empty string. Non-printable or meta characters cannot be entered as new-value settings, so this presents no issues for encoding.

parameter-format Setting	Description
bcd	In ISUP speak, BCD refers to the binary form of the number used as a half a byte nibble, with the byte's lower nibble containing the first digit and the higher containing the second digit. For example, the number 127 is encoded as the two binary bytes 0x2107 on the wire. Using this mode, the Net-Net SBC treats the binary ISUP content as BCD; it should decode it from 0x2107 to the string 1270, and from a string of 127 it should decode it as 0x2107. Since a byte has two nibbles, a nibble might have to be added. And when the Net-Net SBC performs decoding, it cannot know that a BCD byte represents one or two ASCII digits—so it assumes there are two. The number-param setting decodes the parameter as a common number parameter. The Net-Net SBC sees the odd/even bit as in the first bite as telling it how many nibbles to decode correctly, and it will set the odd-even when it decodes. Non-binary digit characters fail to match against the body part if they are contained in the match value, and non-binary characters in the new value results in no operation being performed.
number-param	As the decimal value of the specified number type, treats the parameters as a generic number parameter type. For example, a parameter-type 4 acts on the Called Party Number parameter. When the action type is replace or add, the Net-Net SBC automatically sets the parameter's odd-even bit based on the number being inserted in relation to the new-value setting. If the Numbering Plan Indicator bits are 0b001 (ISDN, E.164), then the Net-Net SBC sets the Nature of Address field to 0b0000100 (international). If this number type is added to a non-existent parameter field, then the Numbering Plan Indicator field is 0b0000011 (national number). If this number type is added to a non-existent parameters field, then the Numbering Plan Indicator field will be set to 0b001 (ISDN,E.164) and the Net-Net SBC will also follow the previous rules. Regardless of the action type you set, the string represented for match-value use for this type will be the numbers of the address fields after the BCD coding. There will be a leading plus sign (+) if the Number Plan is 0b001 and the Nature of Address is 0b0000100 ((international); otherwise, there will not be a plus sign (+). If it cannot convert the data field to a number parameter, the Net-Net SBC will return an empty string. And if the new-value is not in digit form or cannot fit in the specified parameter type field, the Net-Net SBC takes no action.

Adding an ISUP Body to a SIP Message

Unlike the MIME manipulation you can use by setting the SIP header rules accordingly, you can add MIME parts to SIP messages using the MIME rules configuration.

You can configure a SIP manipulation to add an ISUP body to a SIP message. and the Net-Net SBC adds them after any SDP parts if they are present. You can add an ISUP body to a SIP message in two ways:

- You can create a **mime-isup-rule** with the **action** type set to **add**, and enter the entire body in string hexadecimal form in the **new-value** parameter.
- You can leave the **new-value** parameter empty at the **mime-isup-rule** level and create an add rule for an **isup-param-rule**.

In this case, the Net-Net SBC creates the corresponding ISUP message based on the **isup-msg-type** value and supply all of the parameters with their default values. Since the **isup-msg-type** takes a list of values as a valid entry, for this case it only uses the first one. However, the Net-Net SBC ignores the **isup-msg-type** value if you set the **new-value** parameter. And the **isup-param-rule**, if configured, overwrite the default value or add a new parameter based on the defined parameter type.

It is also possible that you might supply a **new-value** both at the **mime-isup-rule** level and at the **isup-param-rule** level. If you do, the **new-value** entry from the **mime-isup-rule** is parsed into an ISUP object and the **isup-param-rule** operates on that object.

ACLI Instructions and Examples

This section shows you how to configure MIME ISUP manipulation.

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-manipulation** and press <Enter>. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.


```
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)#
```
4. Type **mime-isup-rules** and press <Enter>.


```
ACMEPACKET(sip-manipulation)# mime-isup-rules
ACMEPACKET(sip-mime-isup-rules)#
```
5. **name**—Enter a name for this MIME ISUP rule. This parameter is required and has no default.
6. **content-type**—Enter the content type for this MIME rule. This value refers to the specific body part in the SIP message body that is to be manipulated. For example, given a SIP Content-Type header with the value **multipart/mixed;boundary=unique-boundary-1**, you would enter a content-type value of **application/sdp** to manipulate specifically on the SDP portion of the SIP message.

To manipulate the SIP preamble or epilogue, enter the keyword **@preamble** or keyword **@epilogue**.
7. **action**—Choose the type of action you want to be performed: **none**, **add**, **delete**, **manipulate**, **store**, **sip-manip**, and **find-replace-all**. These are the same actions you can select when configuring SIP header manipulation. The default is **none**.
8. **comparison-type**—Enter the way that you want body part of the SIP message to be compared. This choice dictates how the Net-Net SBC processes the match rules against the SIP header. the default is **case-sensitive**. The valid values are: **case-sensitive**, **case-insensitive**, **boolean**, **refer-case-sensitive**, **refer-case-insensitive**, and **pattern-rule**.
9. **msg-type**—Enter the SIP message type on which you want the MIME rules to be performed. Valid values are **any**, **request**, and **reply**. The default value is **any**.

10. **methods**—Enter the list of SIP methods to which the MIME rules applies. There is no default for this parameter.
11. **match-value**—Enter the value to match against the body part in the SIP message. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators.
12. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute.
13. **isup-spec**—Specify how the Net-Net SBC is to parse the binary body; valid values are the enumerated type. The values for this parameter are these SIP ISUP formats:
 - **ANSI-2000** (default)—Corresponding to ANSI T1.113-2000
 - **ITU-99**—Corresponding to ITU Q.763
14. **isup-msg-type**—Identify the specific ISUP message types (such as IAM and ACM) on which to operate. The Net-Net SBC uses with the **msg-type** parameter (which identifies the SIP message) in the matching process. You enter values in this parameters as a list of numbers rather than as an enumerated value because of the large number of ISUP message type, and the range is between 0 and 255. There is no default for this parameter.
15. **mime-header**—Enter the value to be used for comparison with the specific header in the body part of the SIP message. There is no default for this parameter.

To configure ISUP parameters rules:

1. Follows Steps 1 through 4 above.
2. Type **isup-parameter-rules** and press <Enter>.


```
ACMEPACKET(sip-mime-isup-rules)# isup-param-rules
ACMEPACKET(sip-isup-param-rules)#
```
3. **name**—Enter a name for this ISUP parameter rule. This parameter is required and has no default.
4. **mime-header**—Enter the value to be used for comparison with the specific header in the body part of the SIP message. There is no default for this parameter.
5. **action**—Choose the type of action you want to be performed: **none**, **add**, **delete**, **manipulate**, **store**, **sip-manip**, and **find-replace-all**. The default is **none**.
6. **comparison-type**—Enter the way that you want the header in the body part of the SIP message to be compared. This choice dictates how the Net-Net SBC processes the match rules against the SIP header. the default is **case-sensitive**. The valid values are: **case-sensitive**, **case-insensitive**, **boolean**, **refer-case-sensitive**, **refer-case-insensitive**, and **pattern-rule**.
7. **match-value**—Enter the value to match against the header in the body part of the SIP message. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators.
8. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute.
9. **parameter-type**—Using ISUP parameter mapping, enter which of the ISUP parameters on which your want to perform manipulation. This parameter takes values between 0 and 255, and you must know the correct ISUP mapping value for your entry. The Net-Net SBC calculates the offset and location of this

parameter in the body. Note that the value returned from the body does not the type or length, only the parameter value. For example, a parameter-type value of 4 acts on the Called Party Number parameter value.

For detailed information, see the [Net-Net SBC MIME ISUP Parameters \(348\)](#) section above.

10. **parameter-format**—Enter how you want to convert specific parameter to a string representation of that value. Valid values for **parameter-format** are: **number-param**, **hex-ascii** (default), **binary-ascii**, **ascii-string**, and **bcd**. Both match and new values are encoded and decoded by the designated parameter-format type. In this regard, the **match-value** decodes the parameters and the **new-value** encodes the ASCII string into the respective binary format.

For detailed information, see the [Net-Net SBC MIME ISUP Parameters \(348\)](#) section above.

11. Save your work.

Configuration Example

This section provides an example of a SIP manipulation configuration that shows MIME rules and MIME ISUP rules.

```

sip-manipulation
  name
  description
  header-rule
    name
    header-name
    action
    comparison-type
    msg-type
    methods
    match-value
    new-value
    element-rule
      name
      parameter-name
      type
      action
      match-val-type
      comparison-type
      match-value
      new-value
  mime-rule
    name
    Content-Type
    action
    comparison-type
    msg-type
    methods
    match-value
    new-value
    mime-header
      name
      mime-header-name
      action
  manip
  headerRule1
  Date
  add
  case-sensitive
  reply
  elemRule1
  header-value
  add
  any
  case-sensitive
  "August 19, 1967"
  mimeRule1
  application/SDP
  manipulate
  case-sensitive
  request
  mimeHeaderRule1
  Content-Disposition
  add

```

```

                                comparison-type      case-sensitive
                                match-value
                                new-value            "signal;
handling=required"

mime-isup-rules
  name
  content-type
  action                                none
  match-value
  msg-type                             any
  comparison-type                      case-sensitive
  new-value
  methods
  isup-spec                           ansi-2000
  isup-msg-types
  mime-header-rules
    name                               mimeHeaderRule1
    mime-header-name                   Content-Disposition
    action                             none
    match-value
    comparison-type                   case-sensitive
    new-value                         "signal;handling=optional"
  isup-param-rules
    name                               isupRule1
    type                              0
    format                            hex-ascii
    action                             add
    comparison-type                   case-sensitive
    match-value
    new-value                         "signal;handling=optional"

```

Expedited Call Leg Release for Preempted Hairpin Calls

When hairpinned calls are ended because of signaling failures (such as a SIP mid-dialog signaling timeout, or an H.323 TCP keepalive failure) on one call leg, the Net-Net SBC deletes both legs' media flows simultaneously by default. In addition, when the first hairpinned call leg is torn down, the second call leg is gracefully released immediately by the Net-Net SBC creating and sending an appropriate signaling message (e.g., BYE for a SIP call or ReleaseComplete for an H.323 call).

This changes the behavior from older releases where previously, the orphaned call leg in the hairpin scenario would be torn down after the **initial guard timer expires**. The disconnect times of the two call legs, as recorded by the accounting application, will be significantly different, due to the initial guard time for the second call leg firing at a later time. You can override the new default and configure the Net-Net SBC to behave as in previous versions by configuring the **dont-terminate-assoc-legs** option in the media manager.

Accounting Considerations

To indicate cases like this, where the second leg of the hairpinned call was preempted, Net-Net SBC includes the following combination of release/termination causes in the CDR:

```
VSA 49: Acct-Terminate-Cause = NAS_REQUEST
```

VSA 62: Acme-Disconnect-Cause = 8

Expedited Media Termination of Calls with Abnormal Signaling Termination

When a remote endpoint does not receive a BYE message from the endpoint hanging up, and/or the Net-Net SBC does not receive a 200 OK message in response to the BYE from the remote endpoint, the remote call leg's media session is not closed gracefully. In such a case, the media session is closed by a media timer expiring.

In a normal call flow, the Net-Net SBC closes the remote call leg when it receives the 200 OK back from the remote endpoint. In the aforementioned non-standard case, the Net-Net SBC never receives this 200OK. If such situations are expected, you can configure the Net-Net SBC to close the remote call leg's media session as it forwards the BYE message to the remote endpoint, and not wait any further. This is accomplished by configuring the **release media at bye** option in the media manager.

Accounting Considerations

Even if the endpoint that has been hung up on does not respond immediately, the CDR created for the call must have the stop time reflect when the BYE was received by the Net-Net SBC, not when the response was received or when the SBC timed out the abandoned call leg.

To indicate cases like this, where the call termination cannot be completed by standard means, the Net-Net SBC includes the following combination of release/termination causes in the CDR:

```
Acct-Terminate-Cause = Session-Timeout
Acme-Disconnect-Cause = 47
Acme-SIP-Status = 908
```

ACLI Instructions

To configure graceful termination when signaling ended abnormally:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **release-media-at-bye** with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(sip-config)# options +release-media-at-bye
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Introduction

The Net-Net SBC supports H.323 signaling in a way that permits interworking between different H.323 configurations from different providers and carriers. H.323 signaling capabilities on the Net-Net SBC include:

- H.323 V4—Improves on previous versions of the protocol in functionality, scalability, and reliability
- H.225 call signaling with RAS—Establishes connections between H.323 endpoints so real-time data can be exchanged
- H.245—Establishes the type of media flow and manages that flow after it has started
- H.245 tunneling—Encapsulates H.245 messages within H.225/Q.931 messages; when enabled and used with a firewall, one less TCP port is needed for incoming connections
- Fast Start (and Fast Start with parallel H.245)
- H.323 Annex E support for UDP signaling—Provides for multiplexed call signaling over UDP to increase potential call volume and enhance performance

Peering Environment for H.323

The following diagram shows a peering environment for H.323, with the Net-Net SBC positioned between the H.323 core and external providers.



The configuration information shown in the diagram can help you to understand how some basic Net-Net concepts work. The providers in this depiction are configured as realms, and the strategies you see are for session agent group. What you do not see in this diagram is the fact that the Net-Net SBC is configured with sets of H.323 interfaces within it. These interfaces are internal (for an internal provider) and external (for the external providers you see).

In this chapter's [Signaling Modes of Operation \(361\)](#), you will learn how the Net-Net SBC can operate the different modes that support this solution.

Overview

Using H.323 on your Net-Net SBC, you can implement different signaling modes and use features to enhance H.323 capabilities. In the information that follows, you will find detailed explanations of the H.323 signaling mode and of the features available. This chapter gives operational details and later outlines the steps you need to take when features require configuration. Certain H.323 features do not require you to set specific parameters; they are described in this chapter's [H.323 Automatic Features \(409\)](#).

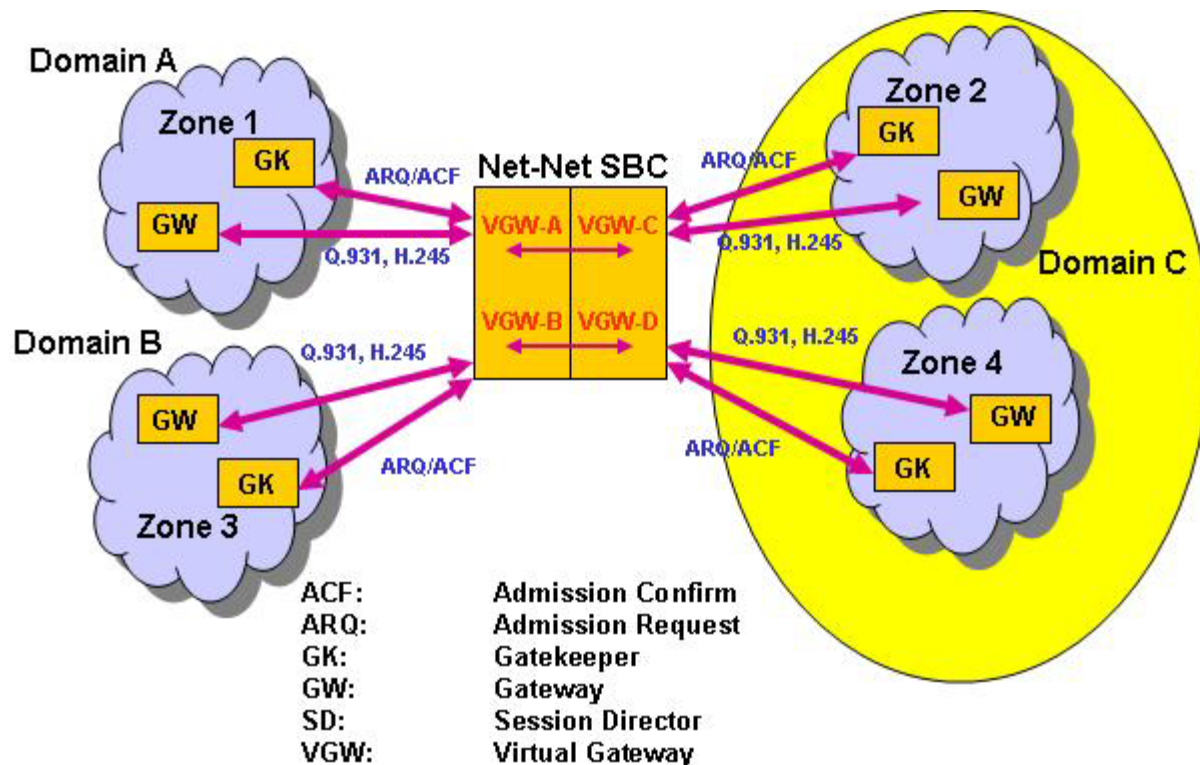
Signaling Modes of Operation

Your Net-Net SBC can operate in different H.323 signaling modes:

- Back-to-back gateway signaling
- Back-to-back gatekeeper proxy and gateway
- Interworking gatekeeper/gateway

Back-to-Back Gateway Signaling

This section explains how signaling takes place when the Net-Net SBC functions as a B2BGW for H.323. The following diagram illustrates the Net-Net SBC acting as a B2BGW.



When configured as a B2BGW, the Net-Net SBC appears as multiple H.323 gateways to multiple networks. You can think of the Net-Net SBC as having virtual gateways, that discovers and registers with a gatekeeper in its respective domain. In this configuration, you need to set the service mode (**isgateway**) parameter for the H.323 interface to **enabled** for two H.323 interfaces. These interfaces are related either through their outgoing interface (**assoc-stack**) parameters or through routing policies.

If you configure your Net-Net SBC to operate in this mode, it does not issue or respond to LRQs by either confirming them or rejecting them.

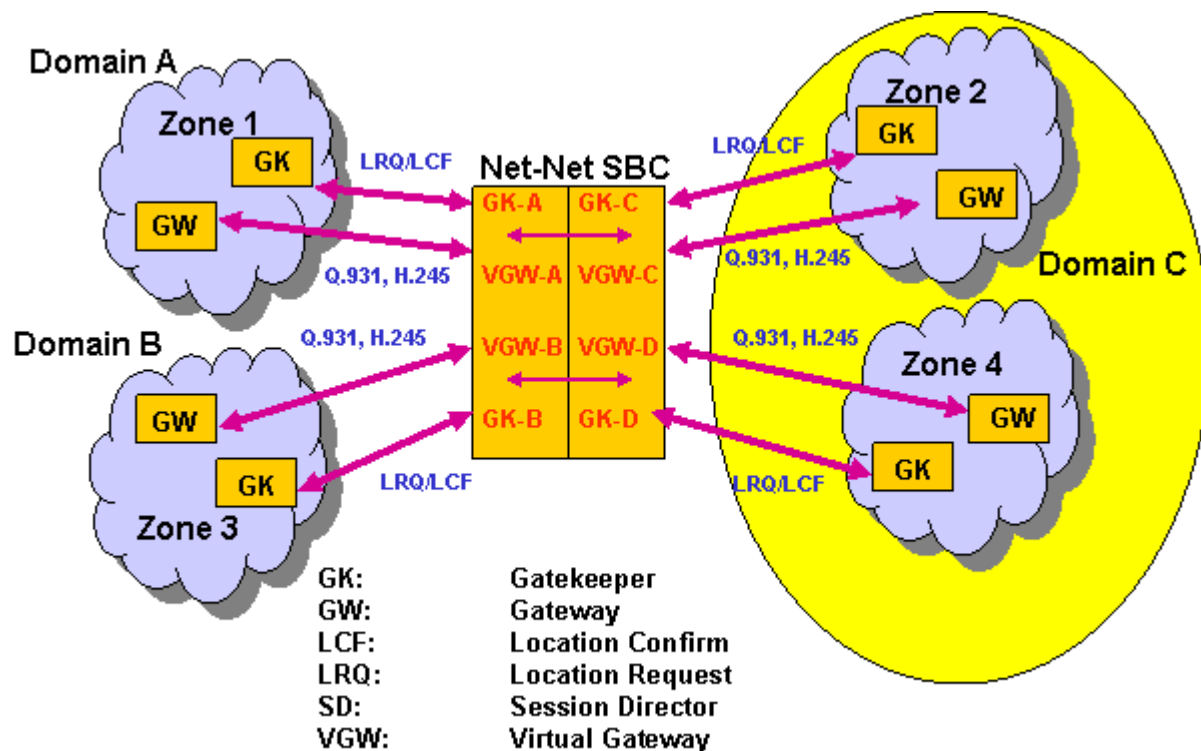
In the diagram above, the Net-Net SBC sends ARQs to the corresponding gatekeeper in its zone when a call is received on the associated interface. In this behavior, the Net-Net SBC acts as a gateway, complying with the H.323 standard, and registers with the configured gatekeeper in its assigned zone. You set all parameters related to the gateway registrations, such as gateway prefix numbers, in the H.323 interface configuration.

In this mode, you can also configure the Net-Net SBC to run like a gateway without a gatekeeper by turning off automatic discovery (auto-gk-discovery) for the remote gatekeeper. When the Net-Net SBC receives a Setup message, it does not send an ARQ and there is no registration for admission requests. Without automatic gateway discovery, the Net-Net SBC uses the local policy to find the appropriate destination for the call. This destination is normally the IPv4 address of the endpoint or gateway, using the well-known port 1720.

If you enable this capability, then the Net-Net SBC finds a gatekeeper.

Back-to-Back Gatekeeper Proxy and Gateway

This section explains how signaling takes place when the Net-Net SBC functions as a back-to-back gatekeeper proxy and gateway for H.323. The following diagram illustrates the Net-Net SBC acting as a B2B gatekeeper proxy and gateway.



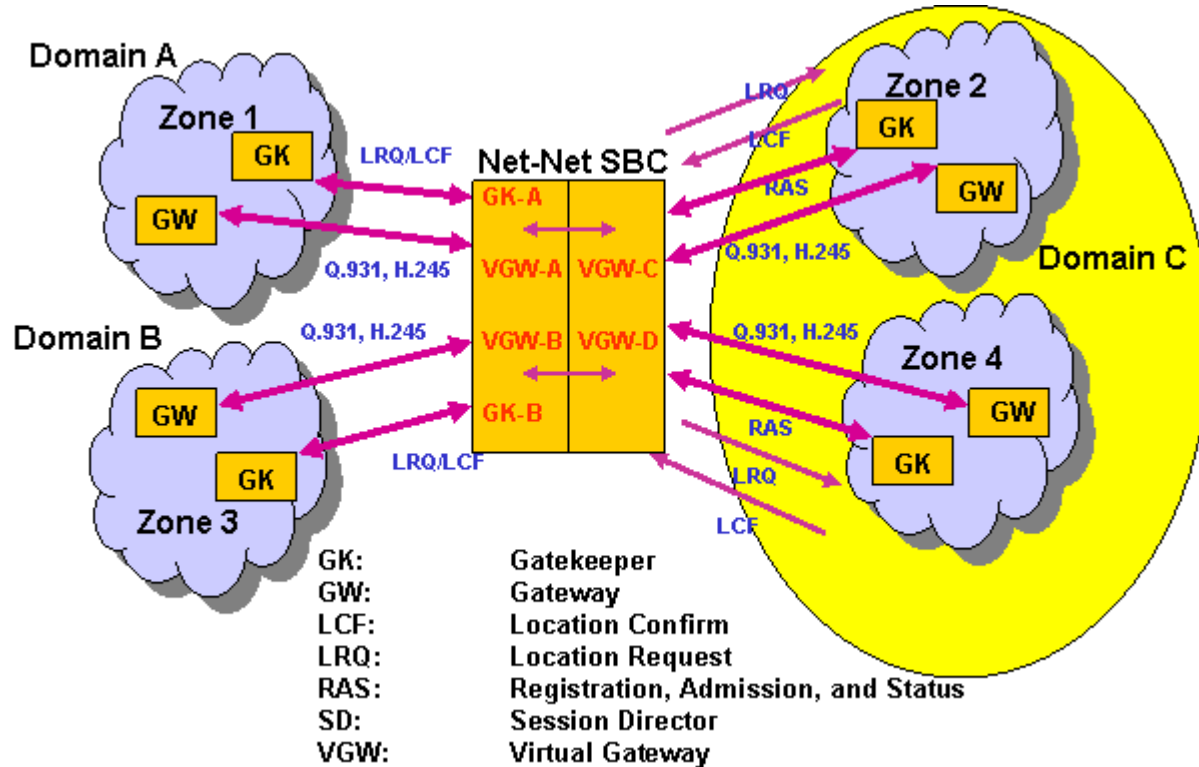
In this application, with the service mode (**isgateway**) parameter set to **disabled**, the Net-Net SBC responds to LRQs and issues LCFs and LRJs. It sends LRQs and LCFs/LRJs to the local IPv4 address for the H.323 interface. The Net-Net SBC responds to the LRQs by providing a signaling address that performs gateway functions.

When you use it as a back-to-back gatekeeper proxy and gateway, the Net-Net SBC does not issue ARQs. In addition, all parameters related to registration, such as gateway prefix numbers, are ignored.

When you do not configure a gatekeeper, the Net-Net SBC uses the local policy to find the appropriate destination for the call. If there is a matching local policy, the Net-Net SBC returns an LCF to the originating gateway. If no local policy matches, the Net-Net SBC rejects the call by sending an LRJ.

Interworking Gatekeeper-Gateway

This section explains how signaling takes place when the Net-Net SBC functions as an interworking gatekeeper-gateway for H.323. The following diagram shows the Net-Net SBC acting as an interworking gatekeeper-gateway.



When you configure your Net-Net SBC for interworking gatekeeper-gateway mode, one H.323 interface behaves as a B2BGW and its associated interface for the corresponding network behaves like a gatekeeper proxy and gateway. The interface for the gatekeeper proxy and gateway issues and responds to LRQ messages on its network. If the Net-Net SBC knows the gatekeeper in the network of the gateway interface (Zone 2), it sends an LRQ to that gatekeeper. If the gatekeeper responds with an LCF or LRJ, the Net-Net SBC forwards it.

If the gatekeeper (in Zone 2) is unknown, then the Net-Net SBC responds to LRQs on the gatekeeper-gateway network (Zone 1) by using the local policy to determine the appropriate destination for the LRQ. If there is no local policy that matches, then the Net-Net SBC sends an LRJ.

For this configuration, the gateway interface has its service mode (**isgateway**) set to **enabled**, and the gatekeeper interface has its service mode (**isgateway**) set to **disabled**.

Realm Bridging with Static and Dynamic Routing

The Net-Net SBC uses static routing and policy-based, dynamic routing to handle H.323 traffic. These types of routing have to do with the way that the outgoing stack is selected.

- Static routing—The incoming H.323 stack always uses the associated H.323 stack that you configure for outgoing traffic; no other stacks are considered.
- Dynamic routing—When there is not an associated stack configured, the Net-Net SBC performs policy-based, dynamic routing known as realm bridging. In this type of realm bridging, the Net-Net SBC checks the configured local policies for address information corresponding to the incoming traffic and finds an address that matches. Next, it checks the next hop in the local policy to determine a realm and uses the first H.323 interface that matches it.

For more information about H.323 and routing, including configuration, refer to this guide's [Session Routing and Load Balancing \(583\)](#) chapter.

Before You Configure

In order to run H.323 on your Net-Net SBC, you need to configure the basic parameters: physical and network interfaces; global system parameters; SNMP, trap receiver, and accounting support, and any holiday information you might want to set.

You should also decide how you want to set up realms and routing (including the use of session agents and session agent groups) to support H.323 operations. For more information about configuring these, refer to the following chapters in this guide:

- [Realms and Nested Realms \(89\)](#)
- [Session Routing and Load Balancing \(583\)](#)

Configuring Global H.323 Settings

When you configure H.323 signaling for your Net-Net SBC, you set global and per-interface parameters. The global parameters govern how the Net-Net SBC carries out general H.323 operations, and these settings are applied to all interfaces you configure for H.323 use. For example, you can turn H.323 support on and off for the entire Net-Net SBC using these settings.

ACLI Instructions and Examples

For the ACLI, global H.323 parameters are:

<code>state</code>	State of the H.323 protocol
<code>log-level</code>	Log level for H.323 stacks
<code>response-tmo</code>	maximum waiting time in sec for response to a SETUP message
<code>connect-tmo</code>	maximum waiting time in sec for establishment of a call
<code>options</code>	optional features/parameters

Accessing Global H.323 Parameters

To access the global H.323 configuration parameters in the ACLI:

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


```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the session-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type **h323-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323-config
```

```
ACMEPACKET(h323-config)#
```

From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a **?** at the system prompt. Access to the H.323 interface (**h323-stack**) configuration also appears.

Global H.323 Settings

To configure global H.323 parameters:

1. **state**—Set the state of H.323 signaling. Leave this parameter set to **enabled** (default) to use H.323 signaling, or change it to **disabled** to disable H.323 signaling.
2. **response-tmo**—Enter the amount of time in seconds that the Net-Net SBC waits between sending a Setup message and tearing it down if there is no response. We recommend you leave this parameter set to 4 (default).

A response might be any of the following messages: Call Proceeding, Connect, or Alerting.
3. **connect-tmo**—Enter the amount of time in seconds that the Net-Net SBC waits between sending a Setup message and tearing it down if it does not specifically receive a Connect message from the endpoint. We recommend that you leave this parameter set to 32 (default).

Receiving a Proceeding or Alert message from the endpoint does not keep this timer from expiring.
4. **options**—Set any options for H.323 features that you want to use. This parameter has a global impact on H.323 behavior, rather than being applied on a per-interface basis. For more information about what parameters you want to configure, refer to the [Options \(406\)](#) section of this chapter.

If you do not configure options for global H.323 behavior, none appears in the configuration display.
5. **log-level**—Select the log level for monitoring H.323 functionality. This parameter overrides the **process-log** level parameter set in the system configuration only for H.323 functionality. If the state parameter in this element is set to disabled, this parameter still overrides the process-log-level field from the system-config element for H.323.

Note that any log level you set here overrides the log level you set in the system configuration's process log level parameter. Available values are:

- EMERGENCY
- CRITICAL
- MAJOR/MINOR
- WARNING
- NOTICE
- INFO
- TRACE
- DEBUG

Configuring H.323 Interfaces

You need to configure H.323 interfaces for inbound and outbound traffic. When you configure H.323 interfaces, you can set:

- Identity and state
- Realm and H.323 interface associations
- H.323 interface settings for the interface's IPv4 address, RAS and Q.931 ports, maximum number of Q.931 ports to allow, and any Annex E support you need
- H.323 system resource allocation

ACLI Instructions and Examples

These are the ACLI parameters that you set:

name	Name of the stack
state	State of the stack
isgateway	Enable the stack to run as a gateway
terminal-alias	List of aliases for terminal
ras-port	Listening port for RAS request
gk-identifier	Gatekeeper's identifier
q931-port	Q.931 call signalling port
alternate-transport	Alternate transport addresses/ports
q931-max-calls	Maximum number of Q.931 calls
max-calls	Stack's maximum number of calls
max-channels	Maximum number of channels per channel

To access the H.323 interface (h323-stack) and service mode parameters:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323-config** and press <Enter>.


```
ACMEPACKET(session-router)# h323-config
```
4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323-config)# h323-stack
```

```
ACMEPACKET(h323-stack)#
```

From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a ? at the system prompt. The display also includes H.323 service mode parameters.

Identity and State

To set the identity, description, and state of the H.323 interface:

1. **name**—Enter a name for the H.323 interface using any combination of characters entered without spaces. For example: InternalGK1.
2. **description**—Enter an optional description for this configuration element. If your entry contains any spaces, you must put the entire entry in double quotation marks.
3. **state**—Enter the state of this H.323 interface by either leaving this parameter **enabled** (default), or setting it to **disabled**.

Realm and Interface Associations

To link this H.323 interface to a realm and to an outgoing H.323 interface:

1. **realm-id**—Enter the identifier for the realm served by this H.323 interface. This parameter must be configured with a valid identifier value from a realm configuration. By default, this parameter is blank.
2. **assoc-stack**—Enter the name of the outgoing H.323 interface that you want to associate with the H.323 interface you are configuring. To use realm bridging with static routing, you need to set the outgoing H.323 interface. If you do not enter a name, the Net-Net SBC uses dynamic, policy-based selection using the local policy. By default, this parameter is blank.

H.323 Signaling Interface Settings

You can set the following parameters to define basic settings for your H.323 interface. This is where you set the IPv4 address for opening sockets, the RAS and Q.931 ports, and the maximum number of Q.931 calls that you want to allow.

This is also where you establish Annex E alternate transport. Annex E supports multiplexed call signaling over UDP so that call volume and performance are potentially enhanced. If you do not configure Annex E support, then this H.323 interface does not listen for Annex E requests.

To configure H.323 interface settings:

1. **local-ip**—Enter the IPv4 address that the H.323 interface uses when opening sockets; this is the default H.323 interface IPv4 address. You must change this parameter from 0.0.0.0 to a valid IPv4 address. For example: 192.168.2.5.
2. **ras-port**—Enter the number of the port on the local IPv4 address (**local-ip**) on which the Net-Net SBC listens for RAS requests. We recommend that you leave this parameter set to the well-known port 1719 (default).

If you leave this parameter set to its default (0), the Net-Net SBC uses a port assigned by the operating system.
3. **q931-port**—Enter the number for the port on the local IP address for the Q.931 call signaling port. We recommend that you leave this parameter set to 1720 (default).
4. **q931-max-calls**—Enter the maximum number of concurrent Q.931 calls you want to allow. You can leave this parameter set to 200 (default). However, this

value should be less than the maximum number of calls you set when configuring [H.323 Features \(370\)](#).

If the number of received Q.931 calls exceeds this number, the H.323 interface returns a *busy* state.

5. **alternate-transport**—Enter a list of one or more Annex E IPv4 address and port combinations for alternate transport. If you do not configure this list, then the Net-Net SBC does not listen for incoming Annex E requests. You must enter the IPv4 address and port combination in the following format, where the two are separated by a colon: **IPv4Address:Port**. By default, this parameter is blank.

H. 323 System Resource Allocation

You can set the following parameters to determine how many concurrent calls and concurrent channels you want to allow for each H.323 interface.

To allocate H.323 system resources:

1. **max-calls**—Enter the maximum number of concurrent calls allowed on this H.323 interface. The default for this parameter is **200**.
2. **max-channels**—Enter the maximum number of concurrent channels allowed for each call associated with this H.323 interface. The default for this parameter is **6**.

The Net-Net SBC checks this parameter on initialization to reserve the appropriate network resources.

Configuring H.323 Service Modes

When you set the H.323 service mode, you configure parameters that define what type of service an H.323 interface provides. These parameters govern how the interface functions when you want it to behave as a gatekeeper or as a gateway.

This is also where you set options that support particular H.323 features for a specific interface. These options are different from the ones you set in the global H.323 configuration because they apply only to the interface where you specify them.

ACLI Instructions and Examples

These are the ACLI parameters that you set:

<code>isgateway</code>	Enable the stack to run as a gateway
<code>registration-ttl</code>	Number of seconds before the registration becomes invalid
<code>terminal-alias</code>	List of aliases for terminal
<code>auto-gk-discovery</code>	Enable automatic gatekeeper discovery
<code>multicast</code>	RAS multicast address
<code>gatekeeper</code>	Gatekeeper's address and port
<code>gk-identifier</code>	Gatekeeper's identifier
<code>h245-tunneling</code>	Enable H.245 Tunneling support
<code>prefixes</code>	List of supported prefixes
<code>process-registration</code>	Enable Registration Request processing
<code>allow-anonymous</code>	allowed requests from H.323 realm

To configure the service mode for the H.323 interface:

1. **allow-anonymous**—Leave this parameter set to **enabled** (default) to accept all H.323 session requests for processing. Set this parameter to **disabled** to accept

requests only from session agents matching the realm identifier set for this H.323 interface.

2. **is-gateway**—To use this interface as an H.323 gateway, leave this parameter set to **enabled** (default). To use this interface as an H.323 gatekeeper, set this parameter to **disabled**.
3. **terminal-alias**—Enter a list of one or more aliases that identify the H.323 interface. This value is either the gateway alias or the gatekeeper identifier, depending on the mode you configure for the interface. The aliases are set in the **sourceInfo** information element of outgoing ARQs. By default, this parameter is blank.

Configuring Gateway Only Settings

If you are using the H.323 interface as a gateway, you might want to set registration time-out and address prefix parameters.

To configure gateway only settings:

1. **registration-ttl**—Enter the number of seconds before a registration becomes invalid or leave the default value of **120**. This value is used during the initial registration process. However, when a registration is confirmed, the time-to-live (TTL) value set by the gatekeeper in the Registration Confirm (RCF) message overrides this value.
2. **prefixes**—Enter a list of prefixes for this H.323 interface. Possible prefix types include: H.323 identifiers, E.164 numbers, URLs, and IPv4 addresses. By default, this parameter is blank.

These prefixes are sent from a gateway interface to a gatekeeper and indicate valid prefixes accepted by that interface for incoming calls. They are used if the interface is configured as a gateway (the **is-gateway** parameter is set to **disabled**).

Your entries for this parameter must appear as they do in the following example:

```
e164=17817566800 url=http://www.acmepacket.com
h323-ID=xyz email=user@acmepacket.com
ipAddress=63.67.143.4:2000
```

Gatekeeper Proxy Settings

If you are using the H.323 stack as a gatekeeper proxy, you might want to set:

- Whether registration processing is enabled or disabled
- Whether or not this H.323 interface is signaling-only
- At what H.225 call stage the H.245 procedures should be initiated

To configure gatekeeper proxy settings:

1. **process-registration**—Leave this parameter set to **disabled** (default) so that the Net-Net SBC drops all RRQs, meaning that it not acknowledges any requests. Set this parameter to **enabled** so that the Net-Net SBC will process any RRQs that arrive on this H.323 interface.

When registration processing is enabled and the Net-Net SBC receives an RRQ on this H.323 interface, it will route the request to the appropriate gatekeeper. After the gatekeeper confirms that registration with an RCF, the Net-Net SBC also confirms it with the endpoint that sent the RRQ. Then the registration becomes part of the Net-Net SBC's registration cache. If this endpoint does not confirm the registration, then the Net-Net SBC will reject the registration with an RRJ and will not cache it.

2. **proxy-mode**—Set this field to either H.225 or H.245 proxy mode that you want to use for the signaling only operation mode. For more information, refer to the [Signaling Only Operation \(387\)](#) description at the beginning of this chapter.

You can leave this field blank (default) if you are not using a proxy mode.

3. **h245-stage**—Set this field to the stage at which the Net-Net SBC transfers the H.245 address to the remote side of the call, or acts on the H.245 address sent by the remote side. Values are: **Setup**, **Alerting**, **Connect** (default), **Proceeding**, **Early**, **noh245**, and **Dynamic**. For more information, refer to the [Dynamic H.245 Stage Support \(383\)](#) description at the beginning of this chapter.

H.323 Features

This section provides general descriptions of the H.323 features available on the Net-Net SBC and instructs you in how to configure them. Not all of the features described in that chapter require configuration.

Fast Start/Slow Start Translations

The Net-Net SBC can translate between Fast Start H.323 endpoints and Slow Start H.323 endpoints. Using this feature, you can reduce delay in establishing media, improve performance, and reduce network congestion caused by a high number of messages being exchanged. Fast Start and Slow Start calls handle information about media for a session in different ways. In a Fast Start call, information about the media is contained in the Setup message. In a Slow Start call, that information is exchanged between endpoints after the session has been established.

When you Fast Start/Slow Start translation, the Net-Net SBC can take a Slow Start call from an H.323 endpoint that does not support Fast Start and re-initiate that call as Fast Start. It also allows an H.323 endpoint that does not support Fast Start to receive a Slow Start call from a Fast Start source because the Net-Net SBC performs all necessary translations.

For the ACLI, the following parameters apply:

fs-in-first-msg	Fast Start must be sent in 1st response to Setup message
call-start-fast	Enable outgoing Fast Start call
call-start-slow	Enable outgoing Slow Start call
media-profiles	List of default media profiles used for outgoing call

Fast Start to Slow Start Translation

The Net-Net SBC supports translations from H.323 Fast Start to Slow Start. Using this feature, an H.323 endpoint that only supports Slow Start can call from a Fast Start source when that call goes through the Net-Net SBC.

In a Fast Start call, the originating H.323 endpoint sends a **fastStart** element in its Setup message. This element contains H.245 OLC messages that allow Fast Start endpoints to establish a media stream when the call is connected. As a result fewer messages are exchanged between the H.323 endpoints than there would be for a Slow Start call (where the **fastStart** element does not appear). Because media information is sent in the Setup request for the session, there is no need to use the media profiles when converting a Fast Start call to Slow Start.

When you enable the slow start option in the H.323 stack configuration, the Net-Net SBC performs Fast Start to Slow Start conversion. During the translation, the Net-Net SBC retains the media information included in the incoming Fast Start

call as it negotiates a connection with the Slow Start endpoint. After a connection with the Slow Start endpoint has been established, the Net-Net SBC negotiates the media capabilities.

Slow Start to Fast Start Translation

When you configure your Net-Net SBC to support H.323 Slow Start to Fast Start translations, you enable an H.323 endpoint that only supports Slow Start to initiate and sustain communication with an H.323 Fast Start endpoint. The Net-Net SBC resolves the Slow Start limitation of exchanging information about media (OLC messages) after the call is connected. The OLC message opens a logical channel, or a unidirectional or bi-directional path used to transmit media packets. Using the Net-Net SBC, you can negotiate the construction of media flows differently, which is described in this section.

When you enable the Fast Start option for calls in the H.323 stack configuration, the Net-Net SBC performs the translation of a Slow Start call into Fast Start. When it receives a Slow Start call, the Net-Net SBC determines its destination and the H.323 stack it uses for the outgoing call.

It is a requirement of this kind of translation that you configure and use media profiles. Since a Slow Start call does not negotiate media until after the call is connected, there needs to be an assumption made about the media to set up a Slow Start to Fast Start call. Media profiles fill this role, and they are assumed to be part of a correct configuration.

The following describes possible scenarios for Slow Start to Fast Start translations.

- When a Slow Start call arrives at the Net-Net SBC and matches one of the session agents that has a media profiles list configured, the outgoing call is set up as a Fast Start call. The session agent's media profiles are used for the logical channels. You must configure the media profiles to reference a codec the endpoint accepts.

If there are no media profiles configured for the session agent, then the Net-Net SBC uses the media profiles list in the H.323 stack configuration to open the logical channels.

- If a Slow Start call arrives at the Net-Net SBC and its destination does not match one of the session agents, the Net-Net SBC uses the media profiles list in the H.323 stack configuration for the outgoing call. If there is a list of media profiles, the outgoing call is set up as a Fast Start call with the media profiles list used to open the logical channels.

If there is no list of media profiles for the outgoing H.323 interface, the Net-Net SBC does not perform Slow Start to Fast Start translation. The Slow Start call exits the Net-Net SBC as it arrived—as a Slow Start call.

- If the egress H.323 interface has the Fast Start option disabled, then the outgoing call uses the Slow Start mode, and the Net-Net SBC does not perform Slow Start to Fast Start translation. In this case, the Slow Start call also exits the Net-Net SBC as it arrived—as a Slow Start call.

Configuration Prerequisites for Slow Start/Fast Start Translations

To perform Fast Start/Slow Start translations, you need to have a standard two-interface configuration already in place.

If you are using the Slow Start to Fast Start translations, you must configure appropriate entries in the media profiles list which is part of the translation parameters. The [Fast Start/Slow Start Translations \(370\)](#) section of the Net-Net Feature chapter describes how the media profiles are used. The list contains the names of media profiles that you configure in the media profile configuration.

Some media profiles are configured by default. If the information you have configured for a media profile collides with the defaults, then your configured ones are loaded. If there are no collisions, then the Net-Net SBC loads the configured and default profiles. The default media profiles are:

Type	Payload	Encoding	Bandwidth
audio	0	PCMU	0
audio	2	G726-32	0
audio	4	G723	0
audio	8	PCMA	0
audio	9	G722	0
audio	15	G728	0
audio	18	G729	0
audio	101	telephone-events	0

Ensure that you use the name of a configured media profile when you enter values in the media profiles list.

ACLI Instructions and Examples

In the ACLI, you can set media profiles that are required for translating H.323 Slow Start to Fast Start. In the ACLI, you set the following:

```

name                encoding name used in sdp rtpmap attribute
media-type          media type used in sdp m lines
payload-type        rtp payload type used in sdp m lines
transport           transport protocol used in sdp rtpmap attribute
req-bandwidth       amount of bandwidth in kilobits required
frames-per-packet   maximum number of frames per packet
parameters          list of <name=value> pairs separated by space
average-rate-limit  average rate limit of rtp flow

```

To configure a media profile:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **media-profile** and press <Enter>.


```
ACMEPACKET(session-router)# media-profile
```

4. **name**—Enter the encoding name used in the SDP `rtmap` attribute. You must enter a name to uniquely identify the media profile, and you will use this value to make lists of media profiles in H.323 interface configurations.
5. **media-type**—Leave this parameter set to `audio` (default), or change it to one of the following as needed: `audio`, `video`, `application`, `data`, `image` or `text`.
6. **payload-type**—Enter the payload type number that corresponds to the encoding name you entered in Step 4. This value identifies the format in the SDP `m` lines. There is no default value for this parameter. To view possible values you might need, refer to this chapter's notes about [About Payload Types \(426\)](#), which includes a table of standard audio and visual encodings. By default, this parameter is blank.

Note: When you use the RTP/AVP transport method, this value must be numeric.

7. **transport**—Leave this parameter set up `RTP/AVP` (default), or change it to the type of transport protocol used in the SDP `rtmap` attribute.
8. **req-bandwidth**—Enter the bandwidth required if it is more than 0 (default).
9. **frames-per-packet**—Enter the maximum number of frames to use per RTP packet up to 256. Leaving this parameters set to 0 (default) means that it is not being used.

The interpretation of this value varies with codec type and with specific codec.

- For frame-based codecs, the frame size is specific to each. For example, a G.729 frame contains ten milliseconds of audio, while a G.723.1 codec frame contains thirty milliseconds.
 - For sample-based codecs such as G.711, each frame contains one millisecond of audio.
10. **parameters**—Enter additional codec information. For example, the G.723.1 codec can have an additional `silenceSuppression` parameter.
 11. **average-rate-limit**—Enter the sustained rate limit in bytes per second for the media flow. Maximum is 125Mbytes per second, and the default is 0.
 12. **peak-rate-limit**—Enter the peak rate for RTP flows in bytes per seconds. The default is 0.
 13. **max-burst-size**—Enter the maximum data size at peak rate in bytes. The default is 0.
 14. **sdp-bandwidth**—Set this parameter to `enabled` to use the AS bandwidth modifier in the SDP in the conditions for the application specific bandwidth modifier. The default is `disabled`.
 15. **sdp-rate-limit-headroom**—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the **average-rate-limit** (rate limit for the RTP flow). The valid entry range is 0 to 100, and the default is 0.

Configuring Fast Start/Slow Start Translations

When you configure an H.323 interface, you configure it for either Fast Start to Slow Start translation or for Slow Start to Fast Start translation. You cannot configure one H.323 interface for both translation modes.

In the ACLI, you will set the following:

<code>fs-in-first-msg</code>	Fast Start must be sent in 1st response to Setup message
<code>call-start-fast</code>	Enable outgoing Fast Start call

<code>call-start-slow</code>	Enable outgoing Slow Start call
<code>media-profiles</code>	list of default media profiles used for outgoing call

ACLI Instructions and Examples

To configure H.323 interfaces for Fast Start/Slow Start translations:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323)# h323-stack
```

```
ACMEPACKET(h323-stack)#
```

From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a **?** at the system prompt. The display also includes H.323 service mode parameters.

5. **fs-in-first-msg**—Set this parameter to **enabled**. Usually, the first message sent is a Proceeding message. If you want to include Fast Start fields in the first message that the Net-Net uses to respond to a Setup message. If you do not want Fast Start fields included, leave this parameter set to **disabled** (default).

6. **call-start-fast**—Leave this parameter to **enabled** if you want Slow Start calls to be translated to Fast Start when this H.323 interface is chosen as the outgoing interface. If this parameter is enabled, **call-start-slow** has to remain disabled.

If you set this parameter set to **disabled** (default), the outgoing call will be set up in the same mode as the incoming call.

7. **call-start-slow**—Set this parameter to **enabled** if you want Fast Start calls to be translated to Slow Start when this H.323 interface is chosen as the outgoing interface. If this parameter is enabled, **call-start-fast** has to remain disabled.

If you leave this parameter set to **disabled** (default), the outgoing call will be set up in the same mode as the incoming call.

8. **media-profiles**—Enter the list of media profiles that you want to use when translating Slow Start calls to Fast Start. This information is used to open logical channels for the outgoing call.

If you enter the name of a media profile that does not exist, the Net-Net SBC will not perform translation. If you leave this parameter empty, the Net-Net SBC will not perform translation.

H.323 Registration Proxy

The Net-Net SBC provides a registration proxy feature that allows a gatekeeper to authenticate a registration before accepting it. This feature is key when two factors are present: authentication is required, and an RRQ from an endpoint includes a token and/or cryptographic token. If authentication for that endpoint is to work, the Net-Net SBC must forward the registration requests received from the endpoint to the gatekeeper separately. When you do not use the H.323 registration proxy, the Net-Net SBC combines all registrations received from H.323 endpoints into a single RRQ and sends it to the gatekeeper. Using the H.323 registration proxy, you can configure the Net-Net SBC to use separate forwarding.

When registration requests are forwarded separately, each RRQ must have a unique CSA. This means that the Net-Net SBC must perform a one-to-one translation of the CSA in the incoming RRQ to a distinct transport address. The translated address replaces the endpoint's CSA in the outgoing RRQ. Then the Net-Net must listen for incoming calls that arrive at this translated transport address for the registered endpoint.

H.235 Authentication Transparency

When operating in this mode, H.235 authentication tokens (cryptotokens) in RAS messages proxied through the Net-Net SBC are passed through transparently.

For applications where Net-Net SBC is between H.323 gateways and a network hosted gatekeeper, the H.235 cryptotokens are passed through unmodified in RAS messages: RRQs, ARQs, and DRQs. This feature allows for secure gateway authentication.

Unique CSA Per Registered Gateway

When operating in this mode, each CSA is mapped to a registered gateway for call routing. The core gatekeeper does not support additive registrations, so a different CSA must be used for each unique registration that goes to the gatekeeper. The gatekeeper does not overwrite previously registered aliases. Also, since the gatekeeper initiates calls to an endpoint on the CSA specified in the RRQ, the Net-Net SBC must listen on the assigned address for incoming calls to that client as long as the client is registered.

Virtual Call Signaling Address

You can configure the Net-Net SBC with:

- A TCP port range for Q.931—Q.931 ports that are frontend ports handled by a real backend socket, and are therefore “virtual”
- ATCP port range for separate H.245 TCP connections—Actual sockets that the Net-Net SBC handles separately

Virtual call signaling address is an H.323 call signaling address that is registered with a gatekeeper, but does not have a corresponding listening socket in the Net-Net SBC. Using the virtual call signaling address means that numerous network transport addresses do not need to be allocated.

Virtual call signaling addresses work by attaching a range of TCP server ports to a single listening TCP socket. After a connection is accepted, the accepting socket created by the server socket operated normally, as though it were created by the server socket that listens on the same transport address as the destination of the arriving packet.

To use virtual call signaling addresses, you specify a Q.931 port range from which the Net-Net SBC can allocate ports. This port range is associated with the virtual call signal IPv4 address you specify. To bind dynamic TCP connections to a port within a port range, you configure a dynamic H.245 port range. The dynamic H.245 port range refers to the separate TCP connection for H.245 that takes place when tunneling is not being used. This enables the Net-Net SBC to select the port to which the TCP socket is bound. These two port ranges cannot overlap.

When a new RRQ has to be forwarded to the gatekeeper, the Net-Net SBC caches the registration and then forwards a modified copy of the RRQ. The Net-Net SBC allocates a virtual call signal address on the gateway stack and uses it to replace the CSA of the registering endpoint in the forwarded RRQ.

Virtual RAS Address

The Net-Net SBC also allocates a virtual RAS address for each endpoint registration. Before forwarding an RRQ from an endpoint, the Net-Net SBC replaces the RAS address of the registering endpoint with the virtual RAS address on the gateway interface.

RAS Message Proxy

When the Net-Net SBC's registration proxy feature is configured, RAS messages to and from endpoints are forwarded, except for the following: GRQ, GCF, GRJ, IRQ, IRR, IACK, and INACK. If the Net-Net SBC receives a valid GRQ on the RAS port of the gatekeeper stack that supports H.323 registration, it responds with a GCF message. Otherwise, it sends a GRJ message.

If the gateway interface receives IRR or IRQ messages, the Net-Net SBC attempts to respond based on the information about the call, and does not forward the messages.

Other RAS messages are forwarded after some modifications:

- Translating the transport address
- Deleting fields that the Net-Net SBC does not support

For further information, about how the Net-Net SBC modifies or deletes RAS message fields, refer to the [About RAS Message Treatment \(428\)](#) section at the end of this chapter.

About Setting Port Ranges

When you configure the H.323 registration proxy feature, you set the Q.931 port range and the dynamic H.245 port range for H.245 connections. If you configure a Q.931 port range, you must also configure a dynamic H.245 port range.

These port ranges cannot overlap because of TCP ports must be unique. The dynamic H.245 port range is used to allocate a real TCP socket, but the Q.931 port range allocates a virtual call signaling address that does not have an associated listening TCP socket.

Note: You should choose these sockets with future Net-Net SBC features about security in mind because future development will support performing admission control based on these port ranges. You will be able to set up filtering rules to allow only inbound packets to configured port ranges.

The following table shows how the Q.931 and dynamic H.245 port ranges work. If you set the start port of 1024 and the number of ports to 1024, you will have configured a port range that starts at 1024 and ends at 2047. So the final port in the range is the start port number added to the number of ports, minus 1. Remember that you cannot overlap the Q.931 and dynamic H.245 port ranges. Notice that the higher the number of the start ports, the fewer ranges of ports you have remaining from which to choose.

Table 1: Registration Proxy Port Ranges

Number of Ports	Start Port	n
1024	1024 * n	1-63
2048	2048 * n	1-31
4096	4096 * n	1-15
8192	8192 * n	1-7
16384	16384 * n	1-3
32768	32768 * n	1

ACLI Instructions and Examples

In the ACLI, the parameters that apply to this feature are:

q931-start-port	Starting port number for port range used for Q.931 call signalling
q931-number-ports	Number of ports in port range used for Q.931 call signalling
dynamic-start-port	Starting port number for port range used for dynamic TCP connections
dynamic-number-ports	Number of ports in port range used for dynamic TCP connections

To configure the H.323 registration proxy:

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```

4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323)# h323-stack
```

```
ACMEPACKET(h323-stack)#
```

5. **q931-start-port**—Enter the number where you want the Q.931 port range to start. To prevent using a well-known reserved port, this number must be greater than or equal to 2048. Valid port values for this parameter are 2000-19999. You must change this value from the default (0).
6. **q931-number-ports**—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. Valid port values for this parameter are 0-17999.

Note: If you have enabled process registration for this H.323 interface, this value must be set to zero because the interface is a gatekeeper that does not support the virtual call signaling address feature.

7. **dynamic-start-port**—Enter the number where you want the dynamic H.245 port range to start. Valid port values for this parameter are 2000-19999.
8. **dynamic-number-ports**—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. Valid port values for this parameter are 0-17999.

H.323 Registration Caching

The Net-Net SBC can cache and proxy an H.225 RRQ between an H.323 endpoint and a gatekeeper. Registration caching has two benefits:

- It allows the aggregation of RRQs sent to a gatekeeper stack and proxies those requests through the gateway stack. If the external gatekeeper associated with the outbound (gateway) interface does not support additive registration, then the Net-Net SBC consolidates the requests by placing them all in the same packet. Otherwise, additive registration is used on the outbound (gateway) interface.
- It allows the gatekeeper stack to use the registration information to route calls from other realms to the endpoints in its realm.

For registration caching, you need to configure at least two H.323 interfaces:

- One gatekeeper interface to receive registrations
- One gateway interface to proxy registrations

The Net-Net SBC caches all successful registrations, using the cache to route calls back to the associated endpoint.

The following diagram shows how RRQs flow during registration caching.



Caveats for Registration Caching

This feature has the following caveats:

- If a gateway stack receives a URQ message from the gatekeeper, it confirms the request with an UCF message. It flushes all registration caching for that stack. However, the Net-Net SBC does not send URQs to the registered endpoints.
- The Net-Net SBC must be rebooted so that the gateway interface can rediscover the gatekeeper under the following circumstances:

Automatic gateway discovery is turned on for the gateway interface by setting the automatic gateway discovery parameter to **enabled**.

Configuration Requirements

For the Net-Net SBC to determine where to route an RRQ, either the associated stack parameter or the gatekeeper identifier field is used.

First, the Net-Net SBC uses the associated interface (assoc-stack) of the gatekeeper interface to find the interface for the outgoing RRQ. If you do not configure an associated interface and the incoming RRQ has a **gatekeeperIdentifier** field, the Net-Net SBC finds a configured gateway interface with a matching **gk-identifier** field and use it as the outgoing interface. If the incoming RRQ does not have a **gatekeeperIdentifier** field and the gatekeeper interface has a configured gatekeeper identifier, the Net-Net SBC finds a gateway interface with a gatekeeper identifier that matches the one set for the gatekeeper interface and then use it as the outgoing interface. If an outgoing interface cannot be determined, the Net-Net SBC rejects the RRQ with the reason **discoveryRequired**.

A configured H.323 interface can be the gateway interface for more than one gatekeeper interface. If a call is received on the gateway interface, the registration cache will be queried to find a registration matching the call's destination. If a registration is found, the interface on which the registration was received will be used as the outgoing interface for the call.

Subsequent ARQ or URQ messages coming from a registered endpoint will be proxied to the gatekeeper using the outgoing gateway interface established during registration. If a registration is not found, an ARJ or a URJ will be sent to the endpoint originating the ARQ or URQ.

A gatekeeper interface can respond to a GRQ if the GRQ is received on its RAS interface. The Net-Net SBC supports GRQ on a multicast address.

ACLI Instructions and Examples

In the ACLI, the parameters that apply to this feature are:

<code>isgateway</code>	Enable the stack to run as a gateway
<code>registration-ttl</code>	Number of seconds before the registration becomes invalid
<code>terminal-alias</code>	List of aliases for terminal
<code>gatekeeper</code>	Gatekeeper's address and port
<code>gk-identifier</code>	Gatekeeper's identifier

To configure the gateway interface parameters for registration caching:

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter> to access the system-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. **isgateway**—Leave this parameter for the service mode set to **enabled** (default).
 Enabling this parameter ensures that registration with the gatekeeper upon startup. It also ensures that all calls will be preceded by an ARQ to the gatekeeper for admission control.
6. **registration-ttl**—Set the registration expiration parameter to the value of the `timeToLive` field in the RRQ sent to the gatekeeper; the default is **120** seconds.
 When the Net-Net SBC receives an RCF from the gatekeeper, it extracts the `timeToLive` field and uses that value as the time interval for keeping the registration of the gateway interface alive. The Net-Net SBC sends a keep-alive RRQ about ten seconds before the registration expires.
 The registration expiration you set value should not be too low because some gatekeepers simply accept the `timeToLive` in the RRQ, resulting in a potentially high volume of RRQs.
7. **terminal-alias**—Set this parameter if the gatekeeper requires at least one terminal alias in an RRQ; by default, this parameter is blank. On startup, the gateway interface registers with the gatekeeper using this terminal alias.
 When the Net-Net SBC forwards an RRQ from an endpoint and if the gatekeeper does not support additive registration, the RRQ has the interface's terminal alias, the aliases of the registering endpoint, and other aliases of all registered endpoints. Otherwise, the RRQ only contains the aliases of the registering endpoint.
8. **gatekeeper** and **gk-identifier**—Configure these parameters if you do not want the Net-Net SBC to perform automatic gatekeeper discovery. If the gatekeeper identifier is empty, then the Net-Net SBC learns the gatekeeper identifier from the `gatekeeperIdentifier` field in the GCF.

Configuring the Gatekeeper Interface for Registration Caching

In the ACLI, the parameters that apply to this feature are:

<code>isgateway</code>	Enable the stack to run as a gateway
<code>gatekeeper</code>	Gatekeeper's address and port
<code>gk-identifier</code>	Gatekeeper's identifier
<code>registration-ttl</code>	Number of seconds before the registration becomes invalid

To configure the gatekeeper interface parameters for registration caching:

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


```
ACMEPACKET# configure terminal
```
2. Type **system** and press <Enter> to access the system-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. **isgateway**—Set this parameter for the service mode to **disabled**.
6. **gatekeeper**—Leave this parameter empty.
7. **auto-gk-discovery**—Set this parameter to **disabled** (default).
8. **gk-identifier**—Set this parameter to the identification of the gatekeeper to which RRQs received on this interface must be proxied. By default, this parameter is blank.
9. **registration-ttl**—Enter the number of seconds to set the `timeToLive` field in the RCF destined for an endpoint. If you do not configure another value, this timer is set to 120 seconds (default).

This value should not be set too high or too low:

- Setting a value that is too high causes the registration to be alive too long. If an endpoint reboots during this interval and re-registers with the same terminal aliases (but changes its call signaling address), the registration will be rejected with the reason `duplicateAlias`.
- Setting a value that is too low puts an unnecessary load on the Net-Net SBC because it has to handle keep-alive registrations from the endpoint constantly, especially when there are many registered endpoints. If an endpoint does not set the `timeToLive` field in its RRQ, the registration of that endpoint will not expire.

If an endpoint registers again without first unregistering itself (e.g., when it crashes and reboots), the Net-Net SBC rejects the registration using the reason `duplicateAlias`. The Net-Net SBC uses this reason when the endpoint's call signaling address (IP address and port) is changed but its terminal aliases remain the same.

**ACLI Registration
Caching Configuration
Example**

In the following example, the H.323 gatekeeper interface (h323-stack) is private and the gateway interface (h323-stack) is public.

```

h323-config
  state                enabled
  log-level            DEBUG
  response-tmo         4
  connect-tmo         32
h323-stack
  name                private
  state              disabled
  realm-id           private
  assoc-stack       public
  local-ip          192.168.200.99
  max-calls         200
  max-channels      4
  registration-ttl  120
  terminal-alias
  prefixes
  ras-port          1719
  auto-gk-discovery disabled
  multicast         0.0.0.0:0
  gatekeeper        0.0.0.0:0
  gk-identifier
  q931-port         1720
  alternate-transport
  q931-max-calls    200
  h245-tunneling   disabled
  fs-in-first-msg  disabled
  call-start-fast  disabled
  call-start-slow  disabled
  media-profiles
  process-registration enabled
  anonymous-connection disabled
  proxy-mode
  filename
h323-stack
  name                public
  state              enabled
  isgateway          enabled
  realm-id           public
  assoc-stack       private
  local-ip          192.168.1.99
  max-calls         200
  max-channels      2
  registration-ttl  120
  terminal-alias
  prefixes
  ras-port          1719
  auto-gk-discovery disabled
  multicast         0.0.0.0:0
  gatekeeper        192.168.1.50:1719
  gk-identifier      gk-public.acme.com
  q931-port         1720

```

alternate-transport	
q931-max-calls	200
h245-tunneling	disabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	

H.245 Stage

The Net-Net SBC allows you to set the earliest stage in an H.323 call when the Net-Net SBC initiates the procedure to establish an H.245 channel for the call. If you have enabled H.245 tunneling by setting the h245-tunneling parameter to **enabled**, then you do not need to configure your system for this feature.

The Net-Net SBC initiates the H.245 procedure by either:

- Sending its H.245 address, or
- Creating a TCP connection to an H.245 address that it has received

You can set this parameter to any of the following stages of an H.323 call: **setup**, **proceeding**, **alerting**, **connect**, **early**, **facility**, **noh245**, and **dynamic**. With the exception of **early**, **noh245**, and **dynamic**, these values correspond to types of H.225/Q.931 messages. The **dynamic** value is described in detail in the next section.

When you configure the **early** value, your Net-Net SBC begins the H.245 procedure at the time the Setup message is sent or received, or when the Connect message is received.

While these values allows for some flexibility about when the H.245 process is started, they are inherently static. All calls in the H.323 stack configuration use the same value, and it cannot be changed from call to call on that stack.

Dynamic H.245 Stage Support

You can configure your Net-Net SBC for dynamic H.245 support, meaning that the point at which the H.245 process begins can be determined dynamically. To support dynamic H.245, the Net-Net SBC sends its H.245 address in the incoming call when it receives an H.245 address in the outgoing call.

Dynamic H.245 Stage for Incoming Calls

When a call comes in on an H.323 interface that you have configured for dynamic H.245 stage support.

The Net-Net SBC includes its H.245 address in the h245Address field of the first H.225/Q.931 message. The Net-Net SBC does this after it receives the first H.225/Q.931 message with an H.245 address in the outgoing call. Based on the first H.225/Q.931 message received by the Net-Net SBC that has an H.245 address, the

Net-Net SBC selects the message in which to include the H.245 address as outlined in the table below.

Message Received with H.245 Address	Message Sent with H.245 Address
Call Proceeding	Call Proceeding, Progress, Alerting, Connect or Facility. The H.245 address is sent in the Call Proceeding message if the Net-Net SBC has not sent a Call Proceeding message in the incoming call. This is true only when you enable the Fast Start in first message parameter for the incoming stack; this parameter establishes whether or not Fast Start information must be sent in the first response to a Setup message. Otherwise, the message in which the H.245 address is sent depends on what message is received after the Call Proceeding message. This is because the Net-Net SBC sends its Call Proceeding message directly after receiving the Setup message.
Progress	Progress
Alerting	Alerting
Connect	Connect
Facility	Facility

When it receives the first H.225/Q.931 message with an H.245 address in the outgoing call, the Net-Net SBC creates a listening socket on the incoming interface. It also includes the socket address and port in the H.245 address of the next H.225/Q.931 message that it sends. If there is no pending H.225/Q.931 message for the Net-Net SBC to send, it instead sends a Facility message with the reason startH245. Then the H.245 channel is established when a TCP connection is made to the listening socket.

For the outgoing leg of a call that came in on the H.323 stack configured for H.245 dynamic stage support, the Net-Net SBC starts establishing the H.245 channel when it receives the first H.225/Q.931 message with H.245 address information. It also starts to establish a TCP connection to the address and port specified in the H.245 address information. The H.245 channel for the outgoing call is established while the H.245 address (h245Address) is sent in the incoming call as described above.

Dynamic H.245 Stage for Outgoing Calls

This section describes what happens when a message exits the Net-Net SBC on an H.323 stack that you have configured for dynamic H.245 stage support.

When the Net-Net SBC receives the first H.225/Q.931 message that has H.245 address information, it establishes an H.245 channel. The Net-Net SBC initiates a TCP connection to the address and port specified in the H.245 address information.

If the incoming call for the session is also on an H.323 stack with dynamic H.245 configured, the Net-Net SBC starts the H.245 procedure in the incoming call. Otherwise, the Net-Net SBC sends its H.245 address in the incoming call based on the H.245 stage support that you have configured.

The process is different when the Net-Net SBC receives a TCS message on the outgoing call before the incoming call reaches its H.245 stage. In this instance, the Net-Net SBC sends a Facility message with the reason startH245 with its H.245 address in order to start the H.245 procedure. The reason is needed in order for the Net-Net SBC to exchange TCS messages with the incoming side of the call.

ACLI Instructions and Examples

To configure H.245 stage support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. **h245-stage**—Set this field to the stage at which the Net-Net SBC transfers the H.245 address to the remote side of the call, or acts on the H.245 address sent by the remote side. Values are: **Setup**, **Alerting**, **Connect** (default), **Proceeding**, **Early**, **noh245**, and **dynamic**.

H.323 HNT

This section explains how H.323 hosted NAT traversal (HNT) works and how to enable this capability on your Net-Net SBC.

The feature enables endpoints behind NATs to originate and terminate calls by resolving the address differences between the NAT and the actual endpoint.

H.323 communication through a NAT becomes an issue when engaging in RAS messaging. While the H.323 standard specifies specific information elements in the RAS messages that indicate the address to which the replies should be sent, these addresses will be behind the NAT and therefore unroutable. The Net-Net SBC solves this problem by sending RAS replies to the layer 3 address from which the associated RAS request was received.

A second issue exists for media channels as the address specified in the H.323 OLC message will be behind the NAT and likewise unroutable. This is resolved by relying on the fact that the forward and reverse channels will utilize the same address and port on the endpoint. By sending media packets to the same address from which the packet are received, media and flow through the NAT.

If you do not use H.323 HNT, the following behavior will occur:

- When an H.323 endpoint is behind a NAT and it registers with a gatekeeper through the Net-Net SBC, the Net-Net SBC tries to send a response back to the endpoint's RAS address rather than to the NAT from which the request was received.
- The same is true for LRQ and IRQ messages because responses without H.323 HNT for outbound sessions, responses were being sent back to the replyAddress or the rasAddress.
- In addition, the Net-Net SBC always induces one-way media because it tries to send the RTP to the media IP address and port it receives in the OLC messages rather than the ephemeral port on the intermediary NAT.

With this ability enabled, however, the Net-Net SBC sends RAS responses back to the address from which the request was received (the NAT). It does not send responses to the endpoint's `rasAddress` or `replyAddress` mentioned in the signaling message. The same is true for RTP. With H.323 HNT for outbound sessions enabled, the Net-Net SBC sends RTP to the IP address and port from which it receives the RTP packets (the NAT).

The call flow below illustrates how this feature works:



Caveats

Keep in mind the following caveats when you are enabling H.323 HNT for outbound sessions on your Net-Net SBC:

- This capability does not apply to calls that require IWF translation between SIP and H.323.

ACLI Instructions and Examples

You can enable this capability for specific H.323 interfaces.

To enable H.323 HNT:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters for the H.323 interface.


```
ACMEPACKET(h323)# h323-stack
```
5. If you are adding this service to a new H.323 interface that you are creating, type **options hnt** (to enable H.323 HNT), and then press <Enter>.


```
ACMEPACKET(h323-stack)# options hnt
```
6. If you are adding this service to an H.323 interface that already exists, type **select** to select the interface to which you want to add the service. Then use the options command and prepend the option with a “plus” (+) sign.
 - If you know the name of the interface, you can type the name of the interface at the **name:** prompt and press <Enter>.
 - If you do not know the name of the interface, press <Enter> at the **name:** prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press <Enter>.
 - If are adding service to an existing interface and you type **options hnt** without a “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

H.323 Party Number-E.164 Support

Some H.323 gateways cannot handle **partyNumber** alias addresses in H.225 messages. The Net-Net SBC lets you convert this address type to **dialedDigits** (E.164). This conversion applies to **sourceAddress**, **destinationAddress**, and **destExtraCallInfo** aliases in Setup messages.

To enable this feature, use the **convertPNTtoE164** value in the **options** field of the H.323 stack configuration.

Signaling Only Operation

When you set the Net-Net to operate in signaling-only mode, it acts like a signaling server. It proxies the call signaling messages between two endpoints. Note, however, that the Net-Net SBC does not function as a RAS proxy; it does not proxy RAS messages.

You have two options for the proxy mode:

- H.245 proxy mode—The Net-Net SBC handles call signaling (H.225) and call control (H.245) messages.
- H.225 proxy mode—The Net-Net SBC handles call signaling

To use this feature, you need to set the proxy mode parameter in the H.323 interface configuration to H.225 or H.245.

H.245

When in H.245 proxy mode, the Net-Net SBC proxies or passes through the call signaling (H.225) messages and the call control (H.245) messages. It allows media to flow between the two H.323 endpoints, as shown in the following diagram.



In some deployments, the media might be treated by a NAT device. When the Net-Net SBC is in H.245 proxy mode, any tunneled H.245 message on the ingress side is tunneled in the egress side. However, if the tunneling is refused on the egress side, a separate H.245 session is established.

H.245 proxy mode support is defined in the following table.

Ingress	Egress
Tunneled	Tunneled
Tunneled	Separate H.245 session
Separate H.245 session	Tunneled
Separate H.245 session	Separate H.245 session

H.225

When in H.225 proxy mode, the Net-Net SBC only proxies call signaling (H.225 messages). The call control (H.245 messages) and the media associated with the session do not go through the Net-Net SBC. Instead, they flow directly between the two H.323 endpoints.

Note: H.225 proxy mode is only used in specific applications and should not be enabled without consultation from your Acme Packet Systems Engineer.

The following diagram shows the flow.



In certain deployments, the call control message and media are exchanged between the two H.323 endpoints themselves. When the Net-Net SBC is in H.225 proxy mode, any tunneled H.245 message on the ingress side is tunneled in the egress side; this is irrespective of the value configured in the value you set for the **h.245-tunneling** parameter in the H.323 stack configuration.

Maintenance Proxy Function

The Net-Net supports a maintenance proxy function for H.323 and enhances the way the Net-Net SBC creates unique RAS ports. You can register endpoints through the Net-Net SBC with unique RAS port. You can also set the H.323 interface on the enterprise side to represent enterprise-side endpoints and thereby register on the carrier side.

The maintenance proxy creates a many-to-one association between the enterprise and the carrier side. Interfaces on the enterprise side can be associated with the carrier side interface, which also must be configured to for the maintenance proxy feature.

You configure the maintenance proxy feature by simply setting an option in the H.323 interface configuration.

ACLI Instructions and Examples

To configure the maintenance proxy function, you need to set two values in the options parameters for the H.323 interface (h323-stack):

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```

4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. **options**—Set the options parameter to **maintenanceProxy**.

Automatic Gatekeeper Discovery

Available only when the H.323 interface is functioning as a gateway, this feature allows for automatic gatekeeper discovery on start-up.

This feature is based on the Net-Net SBC sending a GRQ to the multicast address of the RAS Multicast Group, which is the device group listening on this address. If you do not configure a multicast address, Net-Net SBC uses the well-known address and port 224.0.1.41:1718 in the address-port combination making up this parameter.

Multicast only functions when the Net-Net SBC is discovering an external gatekeeper. The Net-Net SBC does not respond to multicast gatekeeper queries.

When it receives a GCF message from a gatekeeper, the Net-Net SBC registers with the gatekeeper indicated in the GCF. When it receives an GRJ message that contains optional information about alternative gatekeepers, the Net-Net SBC attempts to register with an alternate.

If you do not use automatic gatekeeper discovery, the Net-Net SBC registers with the gatekeeper you configure in the gatekeeper parameter. In this case, the gatekeeper identifier you configure is included in to the RRQ. No registration takes place if you do not establish automatic gatekeeper discovery or if you do not configure the gatekeeper and its identifier.

ACLI Instructions and Examples

To configure automatic gatekeeper discovery:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. **auto-gk-discovery**—Set this parameter to **enabled** to use automatic gatekeeper discovery, or leave it set to **disabled** (default) if you do not want to use this feature.
6. **multicast**—Set this parameter to the address and port where the RAS Multicast Group listens; the default is **0.0.0.0**. If you do not set this parameter, then the Net-Net SBC will use the well-known address and port 240.0.1.41:1718. Your entries in this field will be comprised of an IPv4 address and port values separated by a colon.

H.323 Alternate Routing

You can configure your Net-Net SBC to try more possible routes within given time constraints and number of retries.

Without Alternate Routing Enabled

If you do not enable H.323 alternate routing, the Net-Net SBC tries one possible next hop gateway when routing H.323 calls even if the applicable local policy has multiple next hops configured. If that next hop gateway fails (either because it is busy or out of service), the Net-Net SBC relays the failure back to the caller, who hears a busy tone.

In addition, the call will only be routed to the other available next hops if the first one is:

- A session agent that has gone out of service because its constraints have been exceeded
- A session agent that has gone out of service because it failed to respond to a Net-Net SBC Setup request
- A session agent group

With Alternate Routing Enabled

When you enable H.323 Alternate Routing on your Net-Net SBC, you enable the use of the other next hops in addition to the first one. The retry, when the other available next hops are used, is transparent to the caller. However, the number of retries is limited by the value you set for the CLI **connect-tmo** parameter, and this feature works only if there is more than one matching local policy next hop. If there is not more than one match, even if that match is a session agent group, then the call is only attempted once and the caller must retry it.

If the Net-Net SBC receives a Release Complete message before it receives an Alerting message, then it will try the next hop if there are multiple matches. When there is no more than one match, or if the timer or number of retries is exceeded, the Net-Net SBC proxies the most recently received Release Complete message back to the caller.

The following table shows the cause codes and release complete reasons, and either of the two actions the Net-Net SBC takes:

- **Recur**—Means that the Net-Net SBC performs (or continues to perform) alternate routing
- **Proxy**—Means that alternate routing stops, and the Net-Net SBC sends a release complete message back to the caller

H.323 Release Complete Reason	Q.850 Cause Code	Action
No Bandwidth	34—No circuit available	Recur
Gatekeeper Resources	47—Resource unavailable	Recur
Unreachable Destination	3—No route to destination	Recur
Destination Rejection	16—Normal call clearing	Proxy
Invalid Revision	88—Incompatible destination	Recur
No Permission	111—Interworking, unspecified	Recur
Unreachable Gatekeeper	38—Network out of order	Recur
Gateway Resources	42—Switching equipment congestion	Recur
Bad Format Address	28—Invalid number format	Recur
Adaptive Busy	41—Temporary Failure	Recur

H.323 Release Complete Reason	Q.850 Cause Code	Action
In Conference	17—User busy	Proxy
Undefined Reason	31—Normal, unspecified	Recur
Facility Call Deflection	16—Normal, call clearing	Proxy
Security Denied	31—Normal, unspecified	Recur
Called Party Not Registered	20—Subscriber absent	Recur
Caller Not Registered	31—Normal, unspecified	Recur
New Connection Needed	47—Resource Unavailable	Recur
Non Standard Reason	127—Interworking, unspecified	Recur
Replace With Conference Invite	31—Normal, unspecified	Recur
Generic Data Reason	31—Normal, unspecified	Recur
Needed Feature Not Supported	31—Normal, unspecified	Recur
Tunnelled Signaling Rejected	127—Interworking, unspecified	Recur

ACLI Examples and Instructions

This section describes how to enable H.323 alternate routing. There is a new parameter, and the behavior of the pre-existing **response-tmo** and **connect-tmo** parameters change when you enable this feature on your system.

To enable this feature, you need to set the new **alternate-routing** parameter in the global H.323 configuration to **recur**. The other option for this parameter is **proxy**, which means that the Net-Net SBC performs in the way it did prior to Release 4.1, i.e. try only the first matching local policy next hop that it finds.

You configure H.323 alternate for the global H.323 configuration.

To enable H.323 alternate routing:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. **alternate-routing**—To enable H.323 alternate routing, set this parameter to **recur**. If you want to keep the pre-4.1 behavior where the Net-Net SBC only tries one matching local policy next hop, leave this parameter set to **proxy** (default).
5. **response-tmo**—Enter the time in seconds for the response time-out (or T303 timer). This is the amount of time allowed to elapse during which the Net-Net SBC should receive a response to its Setup message. If the first response to the Net-Net SBC's Setup is a callProceeding, then the Net-Net SBC should receive an Alerting or Connect message before this timer (now T303*2) elapses.

The default for this parameter is 4.

6. **connect-tmo**—Enter the time in seconds for the connect time-out (or T301 timer). This is the amount of time allowed to elapse during which the Net-Net SBC should receive a Connect message.

For alternate routing, this parameter is also used to limit the number of next hops that are tried and the length of time they are tried in case the first next hop fails. The call needs to be established before this timer expires; the call will fail after maximum of 5 retries.

The default for this parameter is 32.

H.323 LRQ Alternate Routing

There are networks where the Net-Net SBC is positioned so that it needs to send an H.225 LRQ request to one signaling entity, and then fall back to another signaling entity when there are no resources available on the first. This might be the case when network contain elements that have limited amounts of channels and/or ports.

To handle situations like this one, the Net-Net SBC can be configured for H.323 LRQ alternate routing.

Without this feature enabled, the Net-Net SBC performs H.323 alternate routing for an H.323 call by finding the alternate route for a local policy when the call setup using H.225/Q.931 fails. Some network configurations, however, require that an LRQ message be sent to a gatekeeper prior to call setup in order to request the destination call signaling address—meaning that the Net-Net SBC will release the call if it does not receive an LCF for that LRQ.

With H.323 LRQ alternate routing enabled, the Net-Net SBC can route the call even when it does not receive the LCF.

How It Works

When the Net-Net SBC routes an H.323 call using a local policy and the applicable route specifies gatekeeper/session agent as the next hop, the Net-Net SBC must send that gatekeeper an LRQ to request the destination for the call signaling address. After it sends the LRQ, the Net-Net SBC might receive either an LCF or an LRJ, or it might receive no response at all. Upon failure—either the receipt of an LRJ or no response within a timeout period—the Net-Net SBC tries alternate routes (additional routing policies) until the call is either set up or the routing list ends. For each alternate route, if the next hop is a gatekeeper/session agent, the Net-Net SBC sends an LRQ to the gatekeeper in order to request the destination call signaling address. Otherwise, the Net-Net SBC simply sets up the call.

For a designated period of time, the Net-Net SBC waits for the a response to the LRQ from the gatekeeper. This timeout period is configured by setting two options in the global H.323 configuration: **ras-tmo** (number of seconds the Net-Net SBC waits before retransmitting a RAS message; default is 4) and **maxRasRetries** (maximum number of times the Net-Net SBC retransmits the RAS; default is 1). The Net-Net SBC calculates the LRQ timeout period by multiplying the **ras-tmo** by the **maxRasRetries** and adding one (**ras-tmo** × **maxRasRetries** + 1).

If an out of service session agent is part of a route, the Net-Net SBC skips it when using alternate routing and uses other routes for the policy.

A session agent might go out of service when it exceeds the maximum number of consecutive transaction timeouts to the maximum number of allowable transaction timeouts. Applicable session agent constrain parameter of note are:

- **trans-timeouts**—Maximum number of allowable transaction timeouts (default is 5)

- **ttr-no-response**—Dictates when the SA (Session Agent) should be put back in service after the SA is taken OOS (Out Of Service) because it did not respond to the Net-Net SBC
- **in-service-period**—Amount of time that elapses before a session agent is put back in service after the **ttr-no-response** period has passed

By default, the Net-Net SBC continues to send LRQ messages to a session agent even if the session agent has already sent an LRJ. However, you might want to place a session agent out of service when it has sent a certain number of LRJs; doing so allows alternate routing to take place faster, but this is an optional feature.

To configure an LRJ threshold, you add the **max-lrj** value to an H.323 session agent's **options** parameter; instructions for how to set it and the required syntax appear below. If you do not set this option, then the Net-Net SBC will not put session agents out of service for matters related to LRJs.

If you do set this option (to a non-zero value), then the Net-Net SBC keeps a count of the LRJs received from a session agent. When it receives an LCF from a session agent, the Net-Net SBC resets the counter to zero. This count is used internally only and is not accessible through statistics displays.

If a session agent exceeds the maximum number of LRJs and goes out of service, it remains in that state until the **ttr-no-response** period has passed and it has transitioned through the **in-service-period** time. If the **ttr-no-response** period is zero, then the session agent is never put out of service.

Caveats

The Net-Net SBC does not support H.323 LRQ alternate routing for these scenarios:

- Calls that require translation between SIP and H.323 (IWF calls)
- For pure H.323 calls where the ingress H.323 interface (stack) is associated with another H.323 interface (stack) that has a valid gatekeeper defined; if there is no valid gatekeeper for the egress interface (stack), this feature may apply

ACLI Instructions and Examples

There is no configuration for H.323 LRQ alternate routing; it is enabled by default. You do, however, need to set the **ras-tmo** and **maxRasRetries** options to set the timeout period.

If you want to set a maximum number of consecutive LRJs to be received from a session agent, you need to add the **max-lrj** value to an H.323 session agent's **options** parameter.

To configure the number of seconds before the Net-Net SBC retransmits a RAS message:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **ras-tmo=x** (where X is number of the seconds that the Net-Net SBC waits

before retransmitting a RAS message; default is 4) with a “plus” sign in front of it, and then press <Enter>.

Set the `maxRasRetries` option in the same way; here, X is the maximum number of times the Net-Net SBC retransmits the RAS; default is 1).

```
ACMEPACKET(h323-stack)# options +ras-tmo=6
```

```
ACMEPACKET(h323-stack)# options +maxRasRetries=2
```

If you type `options` and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the `h323` configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

To limit the number of LRJs received from an H.323 session agent before putting it out of service:

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

```
ACMEPACKET# configure terminal
```

2. Type `session-router` and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type `session-agent` and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
```

4. Use the ACLI `select` command so that you can work with the session agent configuration to which you want to add this option.

```
ACMEPACKET(session-agent)# select
```

5. `options`—Set the options parameter by typing `options`, a <Space>, the option name `max-lrj=X` (where X is the maximum number of allowed LRJs) with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(session-agent)# options +max-lrj=3
```

If you type `options max-lrj=X` (without the “plus” sign), you will overwrite any previously configured options. In order to append the new option to the `session-agent`’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

H.323 CAC Release Mechanism

When an OLC message is sent to the Net-Net SBC and there is insufficient bandwidth available, the Net-Net SBC will reject the incoming OLC. Normally, endpoints decide whether they want to send new OLCs or if they want to release the call. Some endpoints in this situation do neither. When communicating with the last of endpoints, it is desirable for the Net-Net SBC to take action.

The Net-Net SBC supports a option in the H.323 interface called `olcRejectTimer`. When this option is enabled and an OLC is rejected, the stack will:

- If there is another media channel open, the Net-Net SBC will behave as if the release mechanism had not been enabled
- If there are no media channels open, the Net-Net SBC starts a timer for 1 second.
 - If the call is released by the endpoint before the timer expires or another OLC is received from the endpoint before the timer expires, the Net-Net SBC stops the timer and follows expected call handling
 - If the timer expires before either of the above responses from the endpoint occur, the Net-Net SBC releases the call.

ACLI Instructions and Examples

To enable the H.323 CAC release mechanism:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. Use the ACLI **select** command so can add this feature to an existing H.323 interface.


```
ACMEPACKET(h323-stack)# select
```
6. Set the **options** parameter by typing **options**, a <Space>, the option name **olcRejectTimer**, and then press <Enter>.


```
ACMEPACKET(h323-stack)# options olcRejectTimer
```
7. If you are adding this service to an H.323 interface that already exists, type **select** to select the interface to which you want to add the service. Then use the options command and prepend the option with a “plus” (+) sign.
 - If you know the name of the interface, you can type the name of the interface at the **name:** prompt and press <Enter>.
 - If you do not know the name of the interface, press <Enter> at the **name:** prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press <Enter>.
 - If are adding service to an existing interface and type in the option without a “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +olcRejectTimer**.

H.323 Bearer-Independent Setup

In Release 4.1, the Net-Net SBC supports a new H.323 option that enables H.323 Bearer-Independent Setup (BIS). When enabled, this feature allows exception to slow-start to fast-start conversion on the Net-Net SBC.

H.323 BIS Disabled

Unless you enable this feature, the Net-Net SBC performs slow-start to fast-start conversion when a call entering the system as slow-start was routed to an outgoing H.323 interface (stack) with **call-fast-start** set to enabled and there is a list of valid media-profiles in the configuration.

H.323 BIS Enabled

There are certain cases in access deployments where the slow-start to fast-start conversion should not be applied. This is the case when the Setup message contains the Bearer Capability information element (IE), which signals BIS.

When you enable this feature and the Net-Net SBC receives an incoming Setup message that does not contain a fastStart field, the Net-Net SBC checks for the BIS in the incoming Setup before it starts to perform the slow-start to fast-start conversion. If it finds the BIS, then it does not perform the conversion.

This feature can be enabled on a global or a per-interface basis, meaning that you can apply it to your system's entire H.323 configuration or you can enable it only for the interfaces where you want it applied.

ACLI Instructions and Examples

This section explains how to add H.323 BIS support to your global H.323 configuration and to specific H.323 interfaces (stacks).

If you set this option on an H.323 interface (stack), you must set it on the interface (stack) that receives the Setup message with BIS in the Bearer Capability IE.

To enable the H.323 BIS feature globally:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the signaling-related configurations.

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)# session-router
```

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

4. Type **options +bearerIndSetup** and press <Enter>.

```
ACMEPACKET(h323-stack)# options +bearerIndSetup
```

If you type **options bearerIndSetup** without the "plus" (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a "plus" sign as shown in the example above.

To enable the H.323 BIS feature for a specific H.323 interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the signaling-related configurations.

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)# session-router
```

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323)# h323-stack
```

```
ACMEPACKET(h323-stack)#
```

5. Select the H.323 stack to which you want to add H.323 BIS support.

```
ACMEPACKET(h323-stack)# select
```

```
<name>:
```

For a list of configured H.323 interfaces (stacks), press <Enter> at the <name>: prompt. Then enter the number corresponding to the interface where you want to apply this feature.

6. Type **options +bearerIndSetup** and press <Enter>.

```
ACMEPACKET(h323-stack)# options +bearerIndSetup
```

If you type **options bearerIndSetup** without the "plus" (+) sign, you will remove any previously configured options. In order to append the new option

to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

TOS Marking for H.323 Signaling

You can configure your Net-Net SBC to perform TOS/DiffServ marking for H.323 signaling packets. This feature enables you to mark H.323 signaling packets so that they receive specific treatment from upstream devices. This feature assists in routing because you can configure the TOS byte inserted in the H.323 packet to mark the traffic for certain destinations. For example, you can prevent unauthorized video transmission through an audio-only session.

The Net-Net SBC also performs TOS/DiffServ marking for media. For more information, refer to this guide’s *Realms and Nested Realms* chapter. The *Realms and Nested Realms* chapter also contains more information about TOS and DiffServ in general. Refer to that chapter for configuration steps for both kinds of TOS/DiffServ marking: media and signaling.

H.323 Codec Fallback

In the global H.323 configuration, you can enable a parameter that allows the Net-Net SBC to renegotiate—or fallback—to the preferred codec used in an incoming terminal capability set (TCS) from the slow-start side of a slow-start to fast-start H.323 call. When enabled, the Net-Net SBC performs this renegotiation when it detects a mismatch between the codec used in the open logical channel (OLC) opened on the fast-start side of the call, and the codec specified by the slow-start side.

Codec Fallback Disabled

With codec fallback disabled, the Net-Net SBC opens a channel using the codec specified by the northbound side. Since the call manager had specified another preferred codec, the result is a codec mismatch leading to a dropped call.

The following diagram shows how codec mismatches end in dropped calls.

**Codec Fallback
Enabled**

With H.323 codec fall back enabled, the Net-Net SBC attempts to use the preferred codec that the slow-start side of the call specifies. The Net-Net SBC determines matching based on the incoming TCS from the slow-start side and the OLC on the egress side. If the codecs do not match, the Net-Net SBC sends an empty TCS on the egress side and closes the logical channels on the outgoing side of the call.

To trigger a new capabilities exchange, the Net-Net SBC forwards the TCS from the ingress side of the call to the egress endpoint. Then the TCS from the egress endpoint is propagated to the ingress endpoint, and the logical channels are opened. The following diagram shows a call scenario using the H.323 codec fallback feature.



ACLI Instructions and Examples

Note that you configure this feature for your global H.323 configuration, so it has an impact on all H.323 traffic on your system.

To enable H.323 codec fallback:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>. The system prompt will change to let you know that you can configure individual


```
ACMEPACKET(session-router)# h323
```
4. **codec-fallback**—If you want to use the H.323 codec fallback feature, change this parameter from **disabled** (default) to **enabled**.

H.323/TCS Media Sample Size Preservation

For H.323 fastStart calls, the Net-Net SBC can be configured to preserve the packetization interval from the called gateway if it differs from the one offered in the Setup message the calling gateway sent.

When this feature is disabled and in accordance with the ITU H.323 recommendation, the Net-Net SBC changes the packetization rate to the one used by the calling gateway if the one offered by the called gateway differs. In the following example, this means that the Net-Net SBC replaces the packetization interval of 20 with 30 before it forwards the Alerting message to the calling gateway.



However, not all H.323 elements comply with the ITU recommendation. Since some network elements do modify the packetization rate in the dataType element, this behavior is now configurable.

When you enable media sample size preservation, the Net-Net SBC allows the packetization rate to be modified and forwards on the modified dataType element to the calling gateway. In the following example, you can see that the Net-Net SBC

forwards the called gateway's Alerting with the packetization interval of 20 despite the fact that the calling gateway's Setup specified 30.



Note that the calling endpoint might or might not work with the modified dataType.

You can enable this feature for the global H.323 configuration so that it applies to all H.323 fastStart calls, or you can enable it on a per-H.323 interface (stack) basis. When you enable this feature for an individual H.323 interface (stack), the Net-Net SBC performs media sample size preservation for calls egressing on that interface.

ACLI Instructions and Examples

This section shows you how to configure media sample size preservation for the global H.323 configuration and for an individual H.323 interface (stack).

To enable media sample size preservation for the global H.323 configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **forwardFSAcceptedDataType** with a “plus” sign in front of it. Then press <Enter>.


```
ACMEPACKET(h323)# options +forwardFSAcceptedDataType
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the **h323** configuration's options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

To enable media sample size preservation for an individual H.323 interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```

4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI **select** command) the one you want to edit.

5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **forwardFSAcceptedDataType** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(h323-stack)# options +forwardFSAcceptedDataType
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the **h323-stack** configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

H.323-TCS: H.245 Support for H.264 and G722.1

The Net-Net SBC supports the H.264 video codec and the G722.1 audio codec. Especially useful for customer video product offerings in which the Net-Net SBC is deployed, this support further allows the Net-Net SBC to increase ease of use by supporting private addressing. Without this feature enabled (the Net-Net SBC’s previous behavior), the Net-Net SBC required deployment for IANA registered IP addresses—despite the fact that IP VPNs allow for RFC 1918 private addressing.

ACLI Instructions and Examples

To enable this feature, you need to set up media profile configurations appropriately. Media profiles now allow you to set the configuration either as “generic video” or “generic audio.”

H.245 provides for defining new capabilities that are described as H.245 generic capabilities (GenericCapability), which the Net-Net SBC now supports using the H.245 GenericCapability structure. H.264 and G.722.1 are the first codecs the Net-Net SBC offers that use this mechanism.

To set a media profile for generic video support:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **media-profile** and press <Enter>.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

4. **name**—Set the name of the generic video media profile to **genericvideo**. By default, this parameter is blank.

5. **media-type**—Set the media type to use for this media profile; for generic video, change this parameter to **video** (the default is **audio**).
6. **payload-type**—Set the payload type to use for the generic video media profile; by default, this parameter is blank.
7. **transport**—Set the transport type to use for the generic video media profile; the default is **RTP/AVP**.
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic video media profile configuration:

```
media-profile
  name                       genericVideo
  media-type                 video
  payload-type               99
  transport                  RTP/AVP
  req-bandwidth              0
  frames-per-packet          0
  parameters
    average-rate-limit       0
    sdp-rate-limit-headroom  0
    sdp-bandwidth            disabled
```

To set a media profile for generic audio support:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **media-profile** and press <Enter>.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **name**—Set the name of the generic audio media profile to **genericAudio**. By default, this parameter is blank.
5. **media-type**—Set the media type to use for this media profile; for generic video, set this parameter to **audio** (default)
6. **payload-type**—Set the payload type to use for the generic audio media profile.
7. **transport**—Set the transport type to use for the generic audio media profile.
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic audio media profile configuration:

```
media-profile
  name                       genericAudio
  media-type                 audio
  payload-type               104
  transport                  RTP/AVP
  req-bandwidth              0
  frames-per-packet          0
  parameters
```

<code>average-rate-limit</code>	<code>0</code>
<code>sdp-rate-limit-headroom</code>	<code>0</code>
<code>sdp-bandwidth</code>	<code>disabled</code>

International Peering with IWF and H.323 Calls

In previous Net-Net SBC releases, SIP to H.323 IWF calls default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option **cpnType=X**, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outgoing calls on the h323-stack.

In addition to SIP to H.323 IWF calls, the **cpnType=X** option is used to set the Q.931 Number Type for H.323-H.323 calls.

The following is a list of possible values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

ACLI Instructions and Examples

You configure this feature as an option in the **h323-stack** configuration.

To configure the cpnType=X option for H323-H323 calls:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **h323-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323)#
```
4. Type **h323-stacks** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration you want to edit.
5. Set the options parameter by typing **options**, a <Space>, the option name **cpnType=x** (where x is the Q.931 Number Type) with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(h323-stack)# options +cpnType=1
```

If you type **options** without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the h323-stack’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

Options

The options parameter in the global H.323 and H.323 interface configurations allows you to establish the use of specific features; most of those features are customer specific.

You should exercise caution when you apply options because of the fact that many of them are for customer-specific applications. Consult with your Acme Packet systems engineering to find out if using a particular option would be an advantage to you.

Under no circumstance do we recommend that you configure options without Acme Packet consultation. There is the chance that you could set an option that might harm an otherwise sound configuration.

Some of the options described below are only applicable to IWF calls. However, you need to establish them in your H.323 configuration.

Global H.323 Options

The following table lists the options that you might want to use in the global H.323 configuration. Again, we recommend that you consult with an Acme Packet systems engineer about your configuration before using any of these options.

Options	Description
NoDynamicMSD	Net-Net SBC forcefully assumes the “master” role for an outgoing call, and the “slave” role for an incoming call.
AllowOLCWoMSD	Net-Net SBC sends OLC before master/slave determination is complete. Causes the Net-Net SBC to be noncompliant with the H.323 recommendation, which does not permit an OLC to be sent prior to MSD completion.
ModifyMediaInAck	Net-Net SBC accepts and propagates changes to media presented in an OLC Ack. • Applies only to Fast Start OLC/OLC Ack messages embedded in H.225/Q.931 messages during call setup. • Causes Net-Net SBC to be noncompliant with the H.323 recommendation, which does not permit media characteristic to be specified in an OLC to be changed in an OLC Ack.
MapG729	Net-Net SBC maps H.245 G.729 to SDP G.729 with Annex B and vice versa. Applicable only to IWF calls.
ColonG729	Net-Net SBC uses the : (colon) instead of the = (equal sign) in the media attribute line a=fmtp:18 annexb=yes/no when mapping H.245 G.729 or SDP G.729 with Annex B. Applicable only to IWF calls.
IwfLRQ	Net-Net SBC sends an INVITE (with no SDP) to a redirect server in response to an incoming LRQ received on an H.323 interface. If a 3xx message with a redirected contact header is returned, the Net-Net SBC will send an LCF in response to the LRQ. Otherwise, it will send an LRJ.

Options	Description
NoG729AnnexB	SDP received by the IWF with H.729 and no FMTP will be mapped to G.729 on the H.323 side of the call. Can also be set in the session agent options parameter.
sameT38Port	Net-Net SBC does not allocate separate ports for audio and T.38. Net-Net SBC will send the same audio port in the OLCAck that it sees in a request mode for T.38 and a new OLC for T.38.
pvtStats	Net-Net SBC includes program value tree (PVT) statistics in the "show h323d" display that are a sum of the PVT statistics for all H.323 interfaces. Used for debugging purposes.
strayARQTimer	Required the syntax "strayARQTimer=x," where x is the number of seconds the Net-Net SBC waits before tearing down an unsuccessful call in the case of stray ARQs.

H.323 Interface Options

The following table lists the options that you might want to use in the configuration H.323 interfaces. Again, we recommend that you consult with an Acme Packet systems engineer about your configuration before using any of these options.

Option	Description
stackAliasWins	Net-Net SBC will replace the sourceAddress of the incoming Setup message with the terminal alias of the egress interface when copying the incoming sourceAddress to the outgoing Setup message.
uniqueRRQRASAddress	Net-Net SBC will generate unique rasAddress for each RRQ that it sends to a gatekeeper in response to an incoming RRQ received on an H.323 interface configured for process registration. The IP address will be the local-ip of the outgoing interface, so the port is the unique portion of the rasAddress.
nonV4AdditiveRRQ	Gatekeeper associated with the H.323 interface support additive registration even though it does not set the additiveRegistration field in the RRQ message. When sending in the additive mode, the H.323 interface only sends with the RRQ new terminal aliases that need to be registered. In non-additive mode, the interface sense all the terminal aliases that have been registered, plus the new aliases.
cachedTerimnalAlias	Net-Net SBC copies the terminal alias(es) of the registered endpoint to the asourceAddress field of the Setup message. Terminal alias(es) are changed after the Net-Net SBC successfully processes an RRQ from the endpoint.
proxySrcInfo	Net-Net SBC copies the srcInfo from the incoming Setup message to the outgoing Setup message. Otherwise, Net-Net SBC uses its own endpointType for the srcInfo field.
noAliasinRCF	Net-Net SBC does not include any terminal alias in the RCF.
forceH245	Net-Net SBC initiates an H.245 connection after the call is connected. Otherwise, Net-Net SBC listens for an H.245 connection to be initiated by a remote endpoint.
useCPNInRAS	Net-Net SBC uses the calling party number (CPN) IE of the incoming call as the srcInfo of a RAS message sent in the outgoing call (such as an ARQ).
maintenanceProxy	Net-Net SBC registers interfaces on the enterprise side with a gatekeeper on the carrier side, and registers endpoints through the Net-Net SBC with a unique rasAddress. Interfaces on the enterprise side are associated with the carrier interfaces; you set this option on the carrier side.

Option	Description
convertPNTToE164	Net-Net SBC converts the address type partyNumber to dialedDigits (E.164). Conversion applies to sourceAddress, destinationAddress, and destExtraCallInfo aliases in Setup messages.
useCalledPNAsDestInfo	Net-Net SBC uses the H.225 called party number IE as the destinationInfo in ARQ and LRQ requests. Since translation rules can be applied to the Called Party Number, the option enables digit normalization for RAS requests. When not used, Net-Net SBC derives the destinationInfo field in RAS requests from the DestinationAddress field of the incoming Setup.
waitForIncomingH245	On the incoming leg, the Net-Net SBC does not send out its h245Address, but waits for the calling endpoint to send its H245Address. Applies to the outgoing call leg as well: The Net-Net SBC does not send out a Facility with startH245 reason and waits for the called endpoint to send its H245Address.
uniqueRRQSrcPort	Enables H.323 RAS Port Mapping. The Net-Net SBC uses the RAS port that it assigned in the rasAddress parameters of an RRQ message as the UDP source port of the outgoing RRQ. Because this feature is linked to the unique RRQ functionality, be aware of the following before you enable the feature: - Enabling H.323 RAS Port Mapping automatically enables the Net-Net SBC's unique RRQ functionality, eliminating the need for you to configure the latter as a separate option. - Enabling the unique RRQ functionality (by setting the uniqueRRQRASAddress option) does not automatically enable H.323 RAS Port Mapping.
srcCallSignallingPort	Enables use of the Q.931 port value for the port field in the sourceCallSignalAddress parameter in an H.225 Setup message. Useful for customers who configure a separate H.323 interface (stack) on the core side for each external IP-PBX.

H.323 Automatic Features

This section describes H.323 features that are automatically enabled on your Net-Net system. You do not have to configure special parameters to turn them on. Even though you do not have to turn these features on, this section describes what they do and how they work.

Alias Mapping

Alias mapping permits destination addresses to be modified by a gatekeeper.

When sending an ARQ or an LRQ message to a gatekeeper, the Net-Net SBC sets the `canMapAlias` field in that message to `true`. This setting indicates that the Net-Net SBC accepts modified destination information from the gatekeeper. If the resulting ACF or LCF contains `destinationInfo` and/or `destExtraCallInfo` fields, then the Net-Net SBC copies that information respectively to the `destinationAddress` and `destExtraCallInfo` fields of the Setup message. In addition, if the `destinationInfo` is either type `e164` or type `partyNumber`, the Net-Net SBC copies the information into the `calledPartyNumber` information element (IE) of the Setup message, replacing the existing `calledPartyNumber` IE.

You do not need to configure special parameters for this feature; it is enabled automatically.

Call Hold and Transfer

The Net-Net SBC's H.323 call hold and transfer feature supports consultation in addition to call holder and transfer. This feature uses signaling procedures based on the ITU-T recommendations/H.323 specification for what it calls *third party initiated pause and rerouting*.

The following diagram shows how the Net-Net SBC is positioned to provide call hold and transfer support for H.323.



**Call Hold and Transfer:
Basic Call**

The following diagram show the signaling and media flows between the IP PBX and a softswitch. Note how the Net-Net SBC is position to mediate flows between the two devices.

In the Call Proceeding messages forwarded to the IP PBX, the Net-Net SBC uses a non-zero value to ensure that the IP PBX initiates an H.245 session. A progress indicator does not need to be included if the H.245 address is present in any of the following message types: Alerting, Progress, or Connect.

After the Net-Net SBC receives a Call Proceeding message from the softswitch that contains the H.245 address, the Net-Net SBC sends another Call Proceeding with its own H.245 address.

In the following call flow, the softswitch generates message to the gateway. These messages are:

- Initial Address Message (IAM)
- Address Complete Message (ACM)
- Answer Message (ANM)



**Call Hold and Transfer:
Music on Hold**

The following diagram begins with the condition that IP PBX A is already connected with a gateway, with the Net-Net SBC and the softswitch positioned between the two.

You can see in the call flow where the channels for transporting media are closed, and where the RTP/RTCP is stopped. This creates a pause for the call. With the Net-Net SBC mediating the process, IP PBX A and the softswitch exchange TCS and OLC messages that allow music on hold (MHO) to flow between IP PBX A and the gateway.

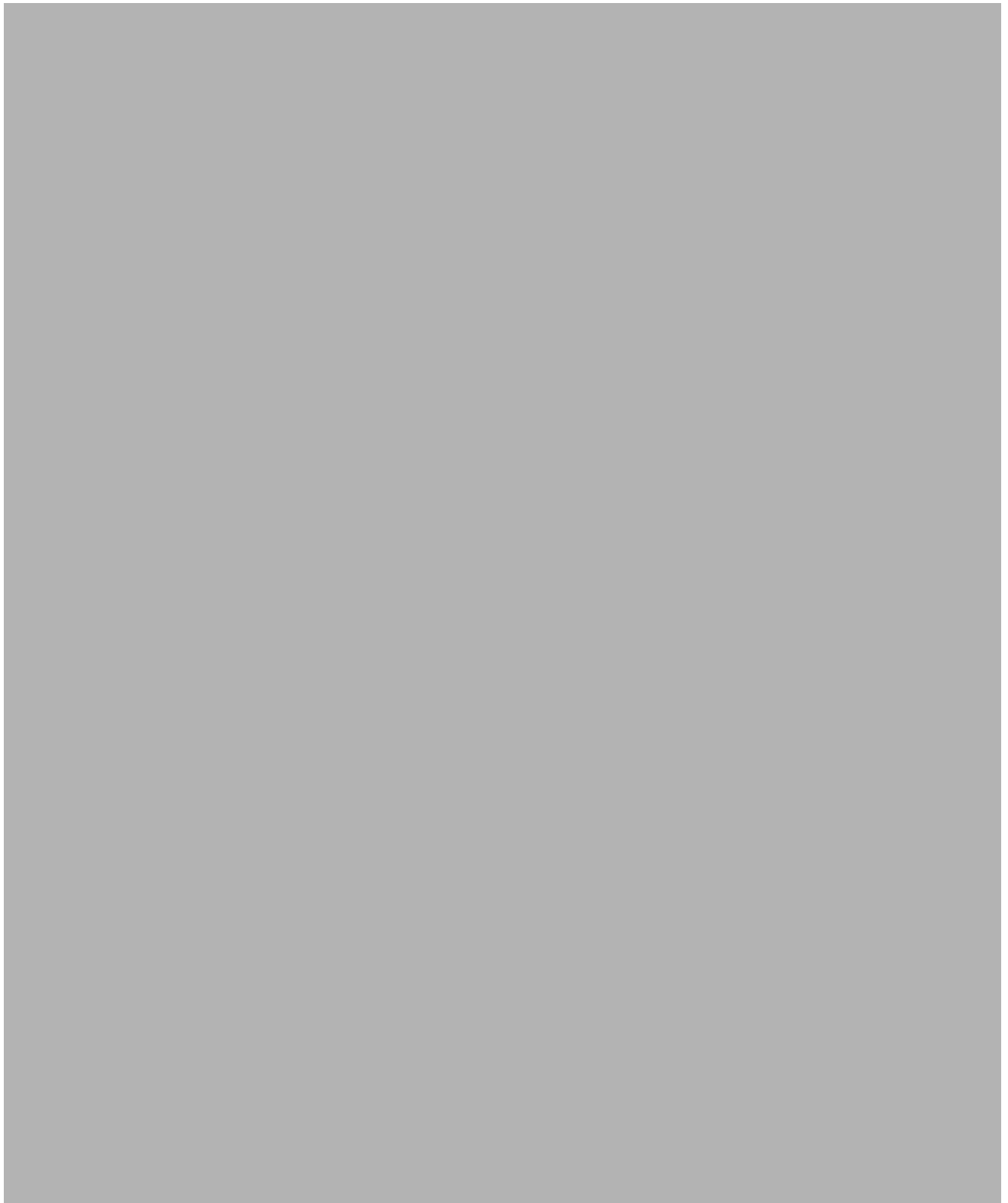


**Call Hold and Transfer:
Transfer**

The following diagram shows how call transfer works on the Net-Net SBC for H.323. In this diagram, you can see:

- Where local ringback occurs
- Where the pause begins and ends
- Where users and gateways are awakened
- Where logical channels are opened and closed







Media Release for SS-FS Calls

When the Net-Net SBC routes a slow-start to fast-start call, it is possible for the same fast-start call to be routed back through the Net-Net SBC making for a hairpin flow. If it does become a hairpin flow, then the Net-Net SBC routes it to its destination as a fast-start to fast-start call. This can result in one-way media if:

- The destination of the hairpin call is in the same realm as the originating slow-start to fast-start call
- The realm reference in the first bullet item is configured to disable in-realm media management
- The called endpoint accepts the proposed fast-start logical channels

The enhancements to the Net-Net SBC's behavior described in this section show how the Net-Net SBC follows additional procedures when setting up a hairpin flow to avoid one-way media when media release occurs.

How It Works

For H.323 calls, the Net-Net SBC establishes media using the H.245 procedures described in the H.245 ITU-T recommendation: control protocol for multimedia communication. It also uses the Fast Connect procedure defined in the H.323 ITU-T recommendation: packet-based multimedia communication systems.

The latter ITU-T recommendation allows a calling endpoint to send a Setup message that contains a fastStart element, a sequence of OLC structures that describe the calling endpoint's proposed forward/reverse logical channels. If the called endpoint accepts this proposal, then logical channels are established.

When the Net-Net SBC translates a call originating in slow-start to fast-start, it uses a Fast Connect procedure in the outgoing leg by sending an outgoing Setup that includes a fastStart element with one or more OLC structures. But when the Net-Net SBC constructs this message, it is unaware of whether the call will become hairpinned or if media release will occur. Because it does not yet have this information, the Net-Net SBC sets the Network Address and the TSAP identifier in the OLC structures to the ingress IP address and port of a corresponding media flow allocated for media traveling between the calling and called endpoints. So if the called endpoint accepts the fastStart the Net-Net SBC proposes, the called endpoint would send its media to the Net-Net SBC. After acceptance, the Net-Net starts H.245 procedures on the slow-start side of the call to set up logical channels on that side. Then the Net-Net SBC updates the IP address and port of the media flows using OLC and OLCAck messages received from the calling endpoint.

This procedure works well for endpoints that are not in the same realm, or that are in the same realm for which media management is disabled, because each endpoint must send its media through the Net-Net SBC. When the endpoints are in the same realm and when media management is enabled, however, the Net-Net SBC must perform additional steps for media release in slow-start to fast-start calls.

To support media release in slow-start to fast-start calls, the Net-Net SBC performs a hold-and-resume procedure on the fast-start side. After it establishes channels on the slow-start side and if it detects media release being enabled, the Net-Net SBC sends an empty TCS to the fast-start side to put that side on hold. Then the called endpoint closes all the logical channels it previously opened in the Fast Connect procedure and stops transmitting to them. And the Net-Net SBC also closes its logical channels. Once the channels are closed, the Net-Net SBC resumes the call by sending a new, restricted TCS to the fast-start side. The restricted TCS only contains the receive and transmit capabilities of the codecs types that the called endpoint accepted in the Fast Connect procedure, and it forces the called endpoint to re-open logical channels of the same codec types accepted in the Fast Connect procedure. Once it receives an OLC from the called endpoint, the Net-Net SBC sends an OLCAck with the Network Address and TSAP identifier for the logical channel from the calling endpoint. Then the Net-Net SBC re-opens logical channels (of the same codec types that it opened in the Fast Connect procedure). If the called endpoint has not changed its Network Address and TSAP identifier for its logical channels, media is re-established after the Net-Net SBC and the called endpoint exit the hold state. The last step is for the Net-Net SBC to re-send the full TCS message from the calling to the called endpoint to inform the called endpoint of the full capabilities of the calling endpoint.

Dependencies

This feature depends on the following assumptions:

- The H.323 endpoint supports the third-party-initiated pause and re-routing feature.
- The H.323 endpoint does not change its Network Address and TSAP identifier when it re-opens the logical channels.
- The H.323 endpoint does not immediately tear down the call when there is not established logical channel in the call.

Hold-and-Resume Procedure

The hold-and-resume procedure has three states:

- **Media Hold**—Starts when the Net-Net SBC sends the empty TCS to the called endpoint to put it on hold.

When it detects media release, the Net-Net SBC puts the called endpoint on hold. It can only do so if it has exchanged the TCS/TCSAck messages and completed master-slave determination with the calling endpoint.

When the Net-Net SBC receives a TCSAck in response to the empty TCS that it sent to the called endpoint, it closes the logical channels it opened as part of the Fast Connect procedure; the called endpoint likewise closes its logical channels. The two then exchange CLC and CLCAck messages, which signals the start of the Media Resume state.

- **Media Resume**—Starts when the Net-Net SBC sends a restricted TCS to resume the call.

The restricted TCS the Net-Net SBC sends contains only the receive/transmit capabilities of the codec types previously accepted by the called endpoint in the Fast Connect procedure. This forces the called endpoint to re-open logical channels of the same codec type that were previously accepted in the Fast Connect procedure.

After sending this TCS, the Net-Net is ready (as specified in the ITU-T recommendations) to take part on the master-slave determination (MSD) process. However, the called party and not the Net-Net SBC initiates the MSD if it is required. The MSD is completed if necessary. Alternately, the called endpoint can start to re-open its logical channels. When it receives the first OLC from the called endpoint, the Net-Net SBC also starts to re-open its logical channels.

- **Media Complete**—Starts when all the logical channels that the Net-Net SBC re-opens are acknowledged by the called endpoint.

When it enters the Media Complete state, the Net-Net SBC updates the called endpoint with the full capabilities of the calling endpoint by sending the full TCS.

H.323 and IWF Call Forwarding

This section describes the Net-Net SBC's H.323 and IWF Call Forwarding feature, which is supported for H.323 calls and for calls initiated in SIP that require interworking to H.323.

Previous Behavior

Prior to Release 4.1, the Net-Net SBC did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

- Fail to release the initial call and initiate the forwarded call
- Drop the entire call when the remote endpoint for the call tore down the session

New Behavior

In the diagram below, you can see that the Net-Net SBC sends the initial Setup message to the gateway, and the gateway returns the Facility message with an alternate address for forwarding. Rather than engaging in its former behavior, the Net-Net SBC now releases the call with the gateway and sends a new Setup to the alternate address from the Facility message.

This new Setup up has no effect on the first call leg, which remains connected.

**How It Works**

When it receives a Facility message with the reason CallForwarded, the Net-Net SBC looks for an alternate transport address in the Facility's alternativeAddress or alternativeAliasAddress element. The Net-Net SBC releases the egress call with the reason facilityCallDeflection. Then it takes one of two courses of action:

- If it does not find an alternative address, the Net-Net SBC releases the ingress call (with the reason facilityCallDeflection).
- If it finds an alternative address and the egress call has not been alerted or answered, the Net-Net SBC at this point tries to initiate a new egress call. The Net-Net SBC uses the alternative alias address to populate the calledPartyNumber information element (IE) and the destination address of the new Setup.

H.323 Sample Call Flow

The following diagram shows how the H.323 Call Forwarding feature works in a purely H.323 environment.

**H.323 NOTIFY Support**

To inform another call party of a certain event or communicate information to it, and H.323 entity might send a NOTIFY message. For example, a gateway might send a NOTIFY message to inform the calling party of a display name for a transferee. In previous releases, the Net-Net SBC did not process such a NOTIFY message, blocking the message from reaching its intended destination.

The Net-Net SBC supports the NOTIFY message so that it can pass through and reach its intended destination.

Caveats

The Net-Net SBC does not support interworking the NOTIFY message to a SIP message for calls that require interworking between H.323 and SIP; this support is for pure H.323 calls only.

ACLI Signaling Mode Configuration Examples

The following ACLI displays provide examples of the [Signaling Modes of Operation \(361\)](#) described earlier in this chapter. Refer to that section to review the details of each mode.

Configuration Fields and Values for B2BGW Signaling

This example provides is a sample for the [Back-to-Back Gateway Signaling \(361\)](#) mode of operation.

```

h323-config
    state enabled
    log-level INFO
    response-tmo 4
    connect-tmo 32

h323-stack
    name zone1
    state enabled
    isgateway enabled
    realm-id zone1realm
    assoc-stack zone2
    local-ip x.x.x.x (IP address of VGW-A)
    max-calls 200
    max-channels 10
    registration-ttl 0
    terminal-alias

    ras-port h323-ID=private
    auto-gk-discovery 1719
    multicast enabled
    gatekeeper 224.0.1.41:1718
    gk-identifier x.x.x.x (IP address of GkZone1)
    q931-port gk-zone1.acme.com
    q931-port 1720
    alternate-transport
    q931-max-calls 200
    h245-tunneling enabled
    fs-in-first-msg disabled
    call-start-fast disabled
    call-start-slow disabled
    media-profiles
    process-registration disabled
    anonymous-connection disabled
    proxy-mode
    filename

h323-stack
    name zone2
    state enabled
    isgateway enabled
    realm-id DomainCrealm
    assoc-stack zone1
    local-ip x.x.x.x(IP address of VGW-C)
    max-calls 200
    max-channels 10
    registration-ttl 0
    terminal-alias

    ras-port h323-ID=acme01
    auto-gk-discovery 1719
    multicast enabled
    gatekeeper 224.0.1.41:1718
    gk-identifier x.x.x.x(IP address of GkZONE2)
    gk-identifier gk-zone2.acme.com

```

q931-port	1720
alternate-transport	
q931-max-calls	200
h245-tunneling	enabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	
h323-stack	
name	zone3
state	enabled
isgateway	enabled
realm-id	zone3realm
assoc-stack	zone4
local-ip	x.x.x.x(IP address of VGW-B)
max-calls	200
max-channels	10
registration-ttl	0
terminal-alias	
	h323-ID=private
ras-port	1719
auto-gk-discovery	enabled
multicast	224.0.1.41:1718
gatekeeper	x.x.x.x(IP address of GkZone3)
gk-identifier	gk-zone3.acme.com
q931-port	1720
alternate-transport	
q931-max-calls	200
h245-tunneling	enabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	
h323-stack	
name	zone4
state	enabled
isgateway	enabled
realm-id	DomainCrealm
assoc-stack	zone3
local-ip	x.x.x.x(IP address of VGW-D)
max-calls	200
max-channels	10
registration-ttl	0
terminal-alias	
	h323-ID=private
ras-port	1719

```

auto-gk-discovery      enabled
multicast              224.0.1.41:1718
gatekeeper             x.x.x.x(IP address of GkZone4)
gk-identifier          gk-zone4.acme.com
q931-port              1720
alternate-transport
q931-max-calls         200
h245-tunneling         enabled
fs-in-first-msg        disabled
call-start-fast        disabled
call-start-slow        disabled
media-profiles
process-registration   disabled
anonymous-connection   disabled
proxy-mode
filename

```

Back-to-Back Gatekeeper Proxy and Gateway

This example provides is a sample for the [Back-to-Back Gatekeeper Proxy and Gateway \(362\)](#) mode of operation.

```

h323-config
  state                enabled
  log-level            INFO
  response-tmo         4
  connect-tmo          32
h323-stack
  name                 zone1
  state                enabled
  isgateway            disabled
  realm-id             zone1realm
  assoc-stack          zone2
  local-ip             x.x.x.x(IP address of VGW-A/GK-A)
  max-calls            200
  max-channels         10
  registration-ttl     0
  terminal-alias
  h323-ID=private
  ras-port             1719
  auto-gk-discovery    disabled
  multicast            0.0.0.0:0
  gatekeeper           x.x.x.x(IP address of GkZone1)
  gk-identifier        gk-zone1.acme.com
  q931-port            1720
  alternate-transport
  q931-max-calls       200
  h245-tunneling       enabled
  fs-in-first-msg      disabled
  call-start-fast      disabled
  call-start-slow      disabled
  media-profiles
  process-registration disabled
  anonymous-connection disabled
  proxy-mode
  filename

```

```

h323-stack
    name                zone2
    state                enabled
    isgateway            disabled
    realm-id             DomainCrealm
    assoc-stack          zone1
    local-ip             x.x.x.x(IP address of VGW-C/GK-C)
    max-calls            200
    max-channels         10
    registration-ttl     0
    terminal-alias

                                h323-ID=acme01
    ras-port             1719
    auto-gk-discovery    disabled
    multicast            0.0.0.0:0
    gatekeeper           x.x.x.x(IP address of GkZONE2)
    gk-identifier        gk-zone2.acme.com
    q931-port            1720
    alternate-transport
    q931-max-calls       200
    h245-tunneling       enabled
    fs-in-first-msg      disabled
    call-start-fast      disabled
    call-start-slow      disabled
    media-profiles
    process-registration disabled
    anonymous-connection disabled
    proxy-mode
    filename

h323-stack
    name                zone3
    state                enabled
    isgateway            disabled
    realm-id             zone3realm
    assoc-stack          zone4
    local-ip             x.x.x.x(IP address of VGW-B/GK-B)
    max-calls            200
    max-channels         10
    registration-ttl     0
    terminal-alias

                                h323-ID=private
    ras-port             1719
    auto-gk-discovery    disabled
    multicast            0.0.0.0:0
    gatekeeper           x.x.x.x(IP address of GkZone3)
    gk-identifier        gk-zone3.acme.com
    q931-port            1720
    alternate-transport
    q931-max-calls       200
    h245-tunneling       enabled
    fs-in-first-msg      disabled
    call-start-fast      disabled
    call-start-slow      disabled
    media-profiles

```

```

process-registration      disabled
anonymous-connection     disabled
proxy-mode
filename
h323-stack
  name                    zone4
  state                   enabled
  isgateway               disabled
  realm-id                DomainCrealm
  assoc-stack             zone3
  local-ip                x.x.x.x(IP address of VGW-D/GK-D)
  max-calls               200
  max-channels            10
  registration-ttl        0
  terminal-alias
  h323-ID=private
  ras-port                1719
  auto-gk-discovery       disabled
  multicast               0.0.0.0:0
  gatekeeper              x.x.x.x(IP address of GkZone4)
  gk-identifier            gk-zone4.acme.com
  alternate-transport
  q931-port               1720
  q931-max-calls          200
  h245-tunneling          enabled
  fs-in-first-msg         disabled
  call-start-fast         disabled
  call-start-slow         disabled
  media-profiles
  process-registration    disabled
  anonymous-connection    disabled
  proxy-mode
  filename

```

Interworking Gatekeeper- Gateway

This example provides is a sample for the [Interworking Gatekeeper-Gateway \(363\)](#) mode of operation.

```

h323-config
  state                   enabled
  log-level               INFO
  response-tmo            4
  connect-tmo             32
h323-stack
  name                    zone1
  state                   enabled
  isgateway               disabled
  realm-id                zone1realm
  assoc-stack             zone2
  local-ip                x.x.x.x(IP address of VGW-A/GK-A)
  max-calls               200
  max-channels            10
  registration-ttl        0
  terminal-alias
  h323-ID=private

```


ras-port	1719
auto-gk-discovery	disabled
multicast	0.0.0.0:0
gatekeeper	x.x.x.x(IP address of GkZone1)
gk-identifier	gk-zone1.acme.com
q931-port	1720
alternate-transport	
q931-max-calls	200
h245-tunneling	enabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	
h323-stack	
name	zone2
state	enabled
isgateway	enabled
realm-id	DomainCrealm
assoc-stack	zone1
local-ip	x.x.x.x(IP address of VGW-C)
max-calls	200
max-channels	10
registration-ttl	0
terminal-alias	
	h323-ID=acme01
ras-port	1719
auto-gk-discovery	enabled
multicast	0.0.0.0:0
gatekeeper	0.0.0.0:0
gk-identifier	gk-zone2.acme.com
q931-port	1720
alternate-transport	
q931-max-calls	200
h245-tunneling	enabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	
h323-stack	
name	zone3
state	enabled
isgateway	disabled
realm-id	zone3realm
assoc-stack	zone4
local-ip	x.x.x.x(IP address of VGW-B/GK-B)
max-calls	200

max-channels	10
registration-ttl	0
terminal-alias	
	h323-ID=private
ras-port	1719
auto-gk-discovery	disabled
multicast	0.0.0.0:0
gatekeeper	x.x.x.x(IP address of GkZone3)
gk-identifier	gk-zone3.acme.com
q931-port	1720
alternate-transport	
q931-max-calls	200
h245-tunneling	enabled
fs-in-first-msg	disabled
call-start-fast	disabled
call-start-slow	disabled
media-profiles	
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	
h323-stack	
name	zone4
state	enabled
isgateway	enabled
realm-id	DomainCrealm
assoc-stack	zone3
local-ip	x.x.x.x(IP address of VGW-D)
max-calls	200
max-channels	10
registration-ttl	0
terminal-alias	
	h323-ID=private
ras-port	1719
auto-gk-discovery	disabled
multicast	0.0.0.0:0
gatekeeper	x.x.x.x(IP address of GkZone4)
gk-identifier	gk-zone4.acme.com

Additional Information

This section contains detailed tables to use as a reference when you are learning about H.323 features or when you are configuring them.

About Payload Types

You set the payload type when you are configuring a media profile to support [Slow Start to Fast Start Translation \(371\)](#).

When you configure media profiles, you might need set the payload type to identify the format in the SDP m lines. For RTP/AVP, the default transport method of a media profile configuration, this will be the RTP payload type number. Newer codecs have dynamic payload types, which means that they do not have an assigned payload type number.

When you use RTP/AVP as the transport method, you should only set the payload type when there is a standard payload type number for the encoding name; otherwise, leave the payload type blank.

The Net-Net SBC uses the payload type value to determine the encoding type when SDP identifies the standard payload type in the `m` line, but does not include an `a=rtpmap` entry. These are two equivalent SDPs:

```
5. c=IN IP4 192.0.2.4
   m=audio 0 RTP/AVP 0

6. c=IN IP4 192.0.2.4
   m=audio 0 RTP/AVP 0
   a=rtpmap:0 PCMU/8000
```

The first does not include the RTP map entry, but uses the standard payload type of 0. If the Net-Net SBC receives an SDP like the first, it uses the payload type 0 to locate the corresponding media profiles configuration. When an `a=rtpmap` is present, the Net-Net SBC uses the encoding name in the `a=rtpmap` line to find the media profile configuration and does not consider the payload type number.

Payload Types for Standard Audio and Visual Encodings

The following is a table of standard audio and visual payload encodings defined in H. Schulzrinne, GND Fokus, "RTP Profile for Audio and Visual Conferences with Minimal Control," RFC 1890, and in the *RTP Parameters* document in IANA's Directory of Generally Assigned Numbers.

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
0	PCMU	A	8000
1	1016	A	8000
2	G721	A	8000
3	GSM	A	8000
4	G723	A	8000
5	DVI4	A	8000
6	DVI4	A	16000
7	LPC	A	8000
8	PCMA	A	8000
9	G722	A	8000
10	L16	A	44100
11	L16	A	44100
12	QCELP	A	8000
13	reserved	A	
14	MPA	A	90000
15	G728	A	8000
16	DVI4	A	11025
17	DVI4	A	22050
18	G729	A	8000

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
19	reserved	A	
20	unassigned	A	
21	unassigned	A	
22	unassigned	A	
23	unassigned	A	
dyn	GSM-HR	A	8000
dyn	GSM-EFR	A	8000
dyn	L8	A	var.
dyn	RED	A	
dyn	VDVI	A	var.
24	unassigned	V	
25	CeIB	V	90000
26	JPEG	V	90000
27	unassigned	V	
28	nv	V	90000
29	unassigned	V	
30	unassigned	V	
31	H261	V	90000
32	MPV	V	90000
33	MP2T	AV	90000
34	H263	V	90000
35-71	unassigned	?	
72-76	reserved for RTCP conflict avoidance	N/A	N/A
77-95	unassigned	?	
96-127	dynamic	?	
dyn	BT656	V	90000
dyn	H263-1998	V	90000
dyn	MP1S	V	90000
dyn	MP2P	V	90000
dyn	BMPEG	V	90000

About RAS Message Treatment

When you enabled the [H.323 Registration Proxy \(374\)](#), the Net-Net SBC modifies and deletes certain fields as outlines in the table below. The Net-Net SBC sends on any fields that are not listed in this table without modifying or deleting them.

Note: Although the Net-Net SBC forwards a field, it does not always support the feature related to that field.

Field Name	Message	Deleted	Modified	Value Used in Modification
alternateEndpoints	RRQ, URQ, ACF	X		
alternateGatekeeper	RCF, URQ	X		
altGKInfo	RRJ, URJ, DRJ	X		
alternateTransportAddresses	RRQ, ARQ, ACF	X		
callModel	ARQ		X	direct
	ACF		X	gatekeeperRouted
callSignalAddress	RRQ		X	Mapped virtual CSA allocated by the Net-Net SBC for registering the endpoint.
	RCF, ARJ		X	CSA of gatekeeper stack
	URQ		X	If URQ is from an endpoint, endpoint's mapped virtual CSA. If URQ is from a gatekeeper, real CSA of endpoint.
destCallSignalAddress	ARQ, ACF	X		
destinationInfo.transportID	ARQ, ACF	X		
destExtraCallInfo.transportID	ARQ, ACF	X		
discoveryComplete	RRQ		X	TRUE
endpointAlias.transportID	URQ	X		
endpointAliasPattern.Wildcard.transportID	URQ			
featureServerAlias.transportID	RCF	X		
gatekeeperIdentifier	RRQ		X	Gatekeeper identifier of the gateway stack, either configured in the H.323 gateway stack or discovered dynamically.
maintainConnection	RRQ, RCF		X	FALSE
multipleCall	RRQ, RCF		X	FALSE
preGrantedARQ.alternateTransportAddresses	RCF	X		

Field Name	Message	Deleted	Modified	Value Used in Modification
preGrantedARQ.useSpecifiedTransport	RCF	X		
rasAddress	RRQ		X	Mapped virtual RAS address allocated by the Net-Net SBC for registering endpoint
remoteExtentsionAddress.trasnp ortID	ARQ, ACF	X		
srcCallSignalAddress	ARQ	X		
srcInfo.transportID	ARQ	X		
supportedH248Packages	RRQ	X		
supportsAltGK	RRQ	X		
supportedPrefixes.prefic.transpor tID	RCF, URQ	X		
terminalAlias.transportID	RRQ	X		
terminalAliasPattern.wilcard.tran sportID	RRQ	X		
willRespondToIIRR	RCF, ACF	X		
willSupplyUIEs	RRQ, ARQ			
uuiesRequested	ACF			
setup			X	FALSE
callProceeding			X	FALSE
connect			X	FALSE
alerting			X	FALSE
information			X	FALSE
releaseComplete			X	FALSE
facility			X	FALSE
progress			X	FALSE
empty			X	FALSE
....				
status			X	FALSE
statusInquiry			X	FALSE
setupAcknowledge			X	FALSE
notify			X	FALSE

H.323 Release Complete Cause Code Customization

When an endpoint ends a call with a Release Complete message through a session agent to the Net-Net SBC, you can modify the expected Q931 Cause code sent back to the other endpoint. This is only applicable in cases where the Session Agent went out of service or the session agent's constraints were exceeded. This behavior is enabled by adding the sa-oos-code=value option to a session agent. The value is entered as a Q931 code, less than or equal to 127.

To customize release code causes on certain fault from a session agent:

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


```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
```
4. Use the ACLI **select** command so that you can work with the session agent configuration to which you want to add this option.


```
ACMEPACKET(session-agent)# select
```
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **sa-oos-code=X** (where X is a cause code value less than 128) with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(session-agent)# options +sa-oos-code=3
```

If you type **options max-lrj=X** (without the “plus” sign), you will overwrite any previously configured options. In order to append the new option to the **session-agent**’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save your work.

H.323 Switchover Behavior

The Net-Net 9200 replicates a portion of H.323 signaling/state information across its two SPUs. When an SPU switchover occurs, calls are maintained, however the Net-Net 9200 begins a process of gracefully terminating them. Likewise, respective STOP CDRs are generated in order to keep track of the calls’ terminated status.

STOP CDRs

The STOP CDR generated after an SPU switchover uses a combination of the *Acme-Disconnect-Cause* and *Acct-Terminate-Cause* attributes to uniquely identify an SPU switchover as the reason for the dropped call. The values for these attributes appear as follows:

- Acct-Terminate-Cause = 6 (Admin Reset)
- Acme-Disconnect-Cause = 3 (Internal Disconnect)

Graceful Media Stream Termination

Dropping a large number of media streams at one time can place a heavy load on the SD4’s processing capability and can have a negative impact on call processing. In order to reduce the impact on call processing, media streams will be dropped in a graceful, controlled manner.

Every 10 seconds, the Net-Net 9200 drops a fixed number of media streams associated with calls that are active at the time of SPU switchover. The number of streams dropped at the end of each interval is configurable with the *red-max-media* parameter. In addition, the CPU load threshold at which to scale back the drop rate is configurable with the *red-cpu-limit* parameter. Finally, the high CPU load media stream drop rate is configured with the *red-load-max-media*.

H.323 Switchover Configuration

H.323 switchover behavior is configured in the *h323-config*’s *switchover-mode* parameter. You can configure the Net-Net 9200 to perform one of the following four actions after an SPU switchover

- none—switchover, media for calls continues until the flow guard timers expire
- drop media—switchover and drop H.323 calls gracefully
- stop cdr—switchover, do not force any calls to terminate, send STOP CDRs when calls are terminated by UAs
- drop stop—switchover, gracefully terminate media streams, send respective STOP CDRs

ACLI Instructions and Configuration

To configure how H.323 calls are dropped after an SPU switchover:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

From this point, you can configure H.323 parameters. The following parameters should be configured for setting the H.323 switchover behavior.
4. **switchover-mode**—Set the SPU switchover mode for H.323 calls. The default is **none**. The following are valid values.
 - **none**—switchover, immediately terminating all H.323 calls
 - **drop-media**—switchover and drop H.323 calls gracefully
 - **stop-cdr**—switchover, do not force any calls to terminate, send STOP CDRs when calls are terminated by UAs
 - **drop-stop**—switchover, gracefully terminate media streams, send respective STOP CDRs
5. **red-max-media**—Set the maximum number of media streams to drop during a 10-second drop interval. The default is 15. Valid values range from 0 - 100.
6. **red-cpu-limit**—Set the maximum CPU load before media streams are dropped at the lower, red-load-max-media value. The default is 30. Valid values range from 1 - 99.
7. **red-load-max-media**—Set the number of media streams to drop during a 10-second drop interval after the red-cpu-limit has been exceeded. The default is 3. Valid values range from 0 - 100.
8. Save and activate your configuration.

Cause Code Completeness

H.323 CDR Generating Behaviors

By enabling enhanced CDR in the H.323 Config, the following cause codes will be reported for the identified action.

- The Net-Net SBC may reject a call because the applicable transcoding policy does not support the codecs offered by the remote H.323 endpoint. In such a case, when the **enhanced-cdr** option is configured in the H.323 config, the Net-Net SBC will insert cause code 44 in both the h323-disconnect-cause and Acme-

disconnect-cause in the CDR. In addition, the disconnect reason remains blank. When **enhanced-cdr** is not configured, h323-disconnect-cause and Acme-disconnect-cause have a value of 3 in the CDR and the “No Route to Destination” disconnect reason is sent to the endpoint.

- The Net-Net SBC may reject a call because the previous hop is not authorized, due to allow-anonymous settings. In such a case, when the **enhanced-cdr** option is configured in the H.323 config, the Net-Net SBC will insert cause code 21 in both the h323-disconnect-cause and Acme-disconnect-cause in the CDR. In addition, a disconnect reason of “SecurityDenied” will be sent to the caller. When **enhanced-cdr** is not configured, h323-disconnect-cause and Acme-disconnect-cause are left blank in the CDR.
- The Net-Net SBC may reject a call because it is out of bandwidth for the codecs offered by the remote H.323 endpoint. In such a case, when the **enhanced-cdr** option is configured in the H.323 config, the Net-Net SBC will insert cause code 34 in both the h323-disconnect-cause and Acme-disconnect-cause in the CDR. In addition, a blank disconnect reason will be sent to the caller. When **enhanced-cdr** is not configured, h323-disconnect-cause and Acme-disconnect-cause are left blank in the CDR.
- The Net-Net SBC may reject a call because of outbound session constraints being exceeded defined in either the Session Agent or Realm configurations. In such a case when the **enhanced-cdr** option is configured in the H.323 config, the Net-Net SBC will insert cause code 34 in both the h323-disconnect-cause and Acme-disconnect-cause in the CDR. In addition, a blank disconnect reason will be sent to the caller.
- When ingress number translation is applied to an H.323 call, and the **enhanced-cdr** option is configured in the H.323 config:
 - The Calling-Station-Id VSA contains the original calling party’s number.
 - The Called-Station-Id VSA contains the original called party’s number.
 - The Acme-Primary-Routing-Number VSA contains the called party’s number after the inbound translation, the one that is used to do the local policy lookup.
 - The Acme-Egress-Final-Routing-Number VSA contains the called party’s number after the outbound translation.
- When the Net-Net SBC clears a call after a media guard timer expires, and the **enhanced-cdr** option is configured in the H.323 config:
 - The Acme-Disconnect-Cause VSA value is set to 16.
 - The Acme-Disconnect-Initiator VSA value is set to 3.
 - The H323-Disconnect-Cause VSA value is set to 6.
- When the **enhanced-cdr** option is configured in the H.323 config, and the outbound leg of a call is rejected with an LRJ/ARJ, the rasReason of undefinedReason will be mapped to the releaseCompleteReason undefinedReason. Also, the Net-Net SBC will insert cause code 31 in both the h323-disconnect-cause and Acme-disconnect-cause in the CDR.

ACLI Instructions

To enable CDR handling of the above H.323 scenarios:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

```
ACMEPACKET(h323)#
```

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **enhanced-cdr** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(h323-stack)# options +enhanced-cdr
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the **h323** configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

H.323 Generating Behavior for OOS Session Agents

When the Net-Net SBC fails a call because an applicable H.323 session agent goes out of service, you can specify a custom cause code to write to the CDR for the failed call.

When the option **sa-oos-code=x** is configured in the session agent configuration element, the user-defined cause code of x written to the h323-disconnect-cause and Acme-disconnect-cause fields in the CDR. In addition, the user-defined value is sent as the the Q.850 cause code to the H.323 endpoint.

If the next-hop for an H.323 Session is a H.323 session agent which does not respond to the SETUP request, the call will be rejected with a cause code of 102. After 5 consecutive failures, the session agent is set to OOS. On the 6th failed call attempt to the failed session agent, the call is rejected with a cause code of 34. If the **sa-oos-code** option is configured in this case, the user defined value will be used for the 6th and subsequent failed attempt.

ACLI Instructions

To enable CDR handling of out of service session agents in the above H.323 scenarios:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
```

```
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration you want to edit.

4. Set the options parameter by typing **options**, a <Space>, the option name **sa-oos-code=x** (where x is the the Q.850 cause code value that will be used by the SBC to use in both the signaling messages and the CDR when the H.323 SA goes OOS.) with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(session-agent)# options +sa-oos-code=102
```

If you type **options** without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the session agent’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Expedited Call Leg Release for Preempted Hairpin Calls

See [Expedited Call Leg Release for Preempted Hairpin Calls \(355\)](#) for details about this feature.

Introduction

Using the Net-Net SBC's interworking (IWF) function, you can interconnect SIP networks with H.323 networks. Considering the large amount of H.323 deployments already in place and the continuing emergence of SIP in new VoIP deployments, the IWF provides a much-needed solution. SIP providers can maintain a single-protocol backbone while exchanging VoIP sessions with H.323 providers.

The [H.323 Signaling Services \(359\)](#) chapter contains information about the H.323 signaling modes of operation that the Net-Net SBC supports. The following H.323 signaling modes of operation can be used when you use the Net-Net SBC's IWF in an access or a peering solution.

- Back-to-back gateway signaling
- Interworking gatekeeper/gateway

Access Network Application

You can configure your Net-Net SBC so that it provides an access solution for your network. The access solution allows SIP-based hosted communications platforms to be extended to enterprise-based H.323 systems. In the figure below, you can see different types of H.323 signaling modes being interworked with SIP. On the H.323 side, the Net-Net SBC can appear to be a gatekeeper or a gateway, depending on how you configure the H.323 interface. On the SIP side, the Net-Net SBC can appear to be a SIP UA or behave as a virtual gateway.



Networking Peering Application

In the IWF network peering solution, you can see the same network elements at work. However, the H.323 side of this IWF application shows the use of a gatekeeper controlled gateway for Peer Network B. Because this is a peering solution, the SIP side of the Net-Net SBC communicates with the SIP proxy or softswitch in the backbone network rather than with the SIP UA or SIP gateway.



How It Works

The Net-Net SBC supports interworking between SIP and H.323 for H.323 Slow Start and Fast Start calls. In addition to describing IWF sessions when initiated from the H.323 side and from the SIP side (with sample call flows), this section provides information you will need when you configure SIP and H.323.

SIP/H.323 Negotiation: H.323 Fast Start

The Net-Net SBC can perform protocol translations for SIP and H.323 Fast Start, where media capabilities are sent with the Setup request for an H.323 session.

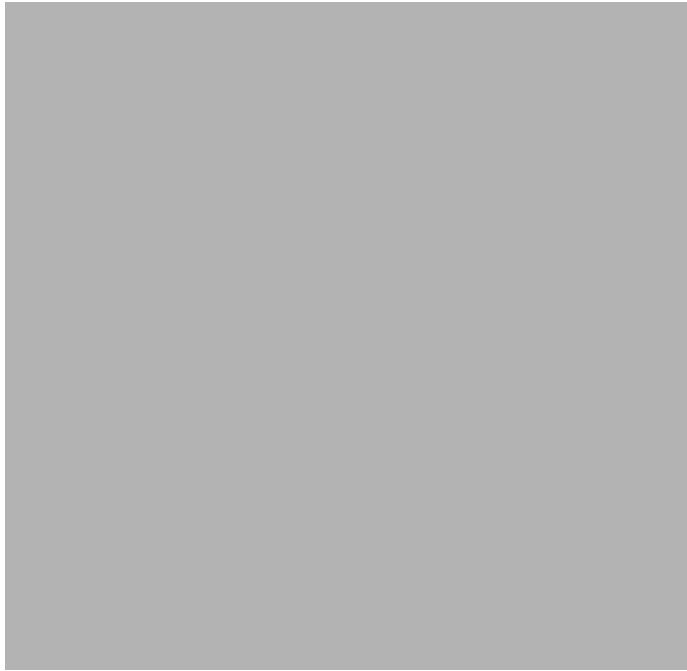
This section's call flow diagrams show how SIP and H.323 messages flow between SIP and H.323 devices, with the Net-Net SBC positioned between the two entities so it can perform translations. The following two sample scenarios with Fast Start appear in the diagrams below, although other scenarios are possible:

- Calls originating in SIP being translated to H.323 Fast Start
- Calls originating in H.323 Fast Start translated to SIP

SIP to Fast Start H.323

In the following diagram below, a SIP endpoint (such as a UA or a SIP Gateway) initiates a session by sending an INVITE message destined for an H.323 endpoint (a GK or GW). Between these entities, the Net-Net SBC is positioned to perform interworking. The Net-Net SBC recognizes that the INVITE message is destined for an H.323 device, and returns a 100 Trying message to the SIP endpoint as it attempts to negotiate the H.323 side of the session. This negotiation starts when the Net-Net SBC initiates the RAS process with the H.323 endpoint by sending either an ARQ or an LRQ, allowing the Net-Net SBC to determine if the H.323 endpoint will accept the session.

Once the H.323 endpoint responds with an ACF or LCF, the Net-Net SBC reissues the SIP INVITE on the H.323 side as an H.225 Setup, which is sent with the OLC. Then the H.323 endpoint responds with Proceeding and Alerting messages (which correspond respectively to SIP 183 Progress and 180 Ringing messages). At that point, the H.323 endpoint sends a Connect message that includes the OpenLogicalChannel message (OLC), announcing the logical channel for media flows has been set up. The Net-Net SBC converts the H.323 OLC to a SIP 200 OK. After receiving the 200 OK, the SIP endpoint sends an ACK, confirming that the session has been established. Because there is no H.323 equivalent for the SIP ACK, the Net-Net SBC does not generate a corresponding message on the H.323 side. At this point, the session is fully established and RTP flows between the endpoints.

**H.323 Fast Start to SIP**

In the diagram below, an H.323 endpoint (a GK or GW) initiates a session by sending a Setup request destined for a SIP endpoint (such as a UA or a SIP Gateway). Between these entities, the Net-Net SBC is positioned to perform interworking. The H.323 endpoint has completed the RAS process prior to sending the SETUP message.

The Net-Net SBC receives the Setup message and then sends a SIP INVITE on the SIP side. The SIP endpoint responds with a 100 Trying; the Net-Net SBC does not resend this message on the H.323 side. Next, the SIP endpoint issues a 180 Ringing message, which the Net-Net SBC reissues to the H.323 endpoint as an Alerting message. The SIP endpoint then sends a 200 OK, retransmitted by the Net-Net SBC as a Connect message that includes an OLC. Once the Net-Net SBC sends an ACK to the SIP endpoint, RTP flows between the endpoints.



SIP/H.323 Negotiation: H.323 Slow Start

The Net-Net SBC can also perform protocol translations for SIP and H.323 Slow Start, where—unlike the cases with Fast Start described above—media information is not sent with the Setup request for an H.323 session. For H.323 Slow Start, media is negotiated after the session is established.

This section's call flow diagrams show how SIP and H.323 messages flow between SIP UA/GW and an H.323 GK/GW, with the Net-Net SBC positioned between the two entities so it can perform translations. Two sample scenarios with Slow Start appear in the diagrams below:

- SIP being interworked to Slow Start H.323
- Slow Start H.323 being interworked to SIP

SIP to H.323 Slow Start

In the following diagram below, a SIP endpoint (such as a UA or a SIP Gateway) initiates a session by sending an INVITE request destined for an H.323 Slow Start endpoint (a GK or GW). Between these entities, the Net-Net SBC is positioned to perform interworking.

The call flow for this type of translation works fundamentally the same way that the translation does for [SIP to Fast Start H.323 \(439\)](#), with the exception of how the media is established. Media is negotiated through the exchange of TCS and OLC messages after the H.323 Connect and SIP 180 Ringing messages have been sent. The first TCS message is sent from the Net-Net SBC to the H.323 endpoint, and it contains information about media capabilities in SDP. The H.323 endpoint accepts and acknowledges this information with a TCS Ack message. Then the H.323 endpoint sends a second TCS, carrying information about the Gateway's capabilities, that the Net-Net SBC accepts and acknowledges. The H.323 endpoint and the Net-Net SBC then exchange OLC and OLC Ack messages that establish the operating mode and Gateway capability. Finally, the Net-Net SBC completes the 200 OK/ACK sequence on the SIP side, and RTP flows between the two endpoints.



H.323 Slow Start to SIP

In the following diagram below, an H.323 endpoint (GW or GK) initiates a session by sending a Setup request destined for a SIP endpoint (such as a UA or a SIP Gateway). Between these entities, the Net-Net SBC is positioned to perform interworking. The H.323 endpoint has completed the RAS process prior to sending the SETUP message.

The call flow for this type of translation works fundamentally the same way that the translation does for [H.323 Fast Start to SIP \(440\)](#), with the exception of how the media is established. When the Net-Net SBC receives an H.323 message destined for a SIP endpoint, it sends a SIP INVITE message that includes default SDP to that SIP endpoint. The default SDP is constructed using information in the media profiles listed for the IWF configuration; if necessary, this media information is amended later in the sequence. Once the call is set up, the Net-Net SBC negotiates media with the H.323 endpoint through a series of TCS/TCS Ack and OLC/OLC Ack messages that establish the operating mode and Gateway capability.

When the Net-Net SBC completes media negotiation with the H.323 endpoint, it issues a re-INVITE to the SIP endpoint that contains the updated information needed for media transmission. In response, the SIP endpoint sends a 200 OK

message that the Net-Net SBC answers with an ACK. Then RTP can flow between the two endpoints.



Status and Codec Mapping

The Net-Net SBC maps SIP and H.323 status codes as described in this section. Status and codec mapping do not require configuration; they occur transparently.

IWF Termination from H.323

When a call that requires the IWF terminates from the H.323 side, the Net-Net SBC uses the mapping scheme in the following table to determine the appropriate SIP status.

H.323 Disconnect Reason	SIP Status
No Bandwidth	480 Temporarily Unavailable
Gatekeeper Resource	404 Not Found
Unreachable Destination	404 Not Found
Destination Rejection	603 Decline

H.323 Disconnect Reason	SIP Status
Invalid Revision	505 Version Not Supported
No Permission	401 Unauthorized
Unreachable Gatekeeper	503 Service Unavailable
Gateway Resource	480 Temporarily Unavailable
Bad Format Request	400 Bad Request
Adaptive Busy	486 Busy Here
In Conference	486 Busy Here
Undefined Reason	500 Internal Server Error
Facility Call Deflection	486 Busy Here
Security Denied	401 Unauthorized
Called Party Not Registered	404 Not Found
Caller Not Registered	401 Unauthorized

IWF Termination During H.323 RAS

When a call that requires the IWF terminates from the H.323 side during RAS and generates an error, the Net-Net SBC uses the mapping scheme in the following table to determine the appropriate SIP status.

H.323 RAS Error	SIP Status
Called Party Not Registered	404 Not Found
Invalid Permission	401 Unauthorized
Request Denied	503 Service Unavailable
Undefined	500 Internal Server Error
Caller Not Registered	401 Unauthorized
Route Call To Gatekeeper	305 User Proxy
Invalid Endpoint ID	500 Internal Server Error
Resource Unavailable	503 Service Unavailable
Security Denial	401 Unauthorized
QoS Control Not Supported	501 Not Implemented
Incomplete Address	484 Address Incomplete
Route Call to SCN	302 Moved Temporarily
Aliases Inconsistent	485 Ambiguous
Not Currently Registered	401 Unauthorized

IWF RAS Registration Failure Code Mapping

For calls that require interworking between H.323 and SIP, the Net-Net SBC supports IWF response code mapping. This feature enables the Net-Net SBC to support configurable SIP response codes for IWF calls that fail during RAS, when the

Net-Net SBC has been unable to register with a gatekeeper; this allows a wider range of more accurate response codes to be communicated.

When this feature is not enabled, the Net-Net SBC generates a 404 Not Found when a SIP-to-H.323 call fails as a result of the stack's failure to register with a gatekeeper.

When the condition noted above takes place, the response code can be any of the ones listed in this table. The code values listed in the table are used to specify the code to which you want to map.

Code	Description
403	Forbidden
406	Not Acceptable
408	Request Timeout
410	Gone
420	Bad Extension
480	Temporarily Unavailable
486	Busy Here
487	Request Terminated
500	Server Internal Error
503	Service Unavailable
504	Server Time-out
600	Busy Everywhere
603	Decline

To enable IWF RAS registration failure code mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**iwfRegFailCode=X**), and then press <Enter>. X is the SIP response code that you want to use; the table above lists the supported response codes that are supported.


```
ACMEPACKET(h323)# options +iwfRegFailCode=503
```

If you type **options iwfRegFailCode=X**, you will overwrite any previously configured options. In order to append the option to the options list, you must prepend the new option with a “plus” sign as shown in the previous example.

IWF Termination from SIP

When an call the requires the IWF terminates from the SIP side, the Net-Net SBC uses the mapping scheme in the following table to determine the appropriate H.323 Release Complete Reason code.

SIP Status	H.323 Release Complete Reason
300 Multiple Choices	Undefined Reason
401 Unauthorized	Security Denied
402 Payment Required	Undefined Reason
403 Forbidden	No Permission
404 Not Found	Unreachable Destination
405 Method Not Allowed	Undefined Reason
406 Not Acceptable	Undefined Reason
407 Proxy Authentication Required	Security Denied
408 Request Timeout	Adaptive Busy
409 Conflict	Undefined Reason
410 Gone	Unreachable Destination
411 Length Required	Undefined Reason
414 Request-URI Too Large	Bad Format Address
415 Unsupported Media Type	Undefined Reason
420 Bad Extension	Bad Format Address
480 Temporarily Unavailable	Adaptive Busy
481 Call/Transaction Does Not Exist	Undefined Reason
482 Loop Detected	Undefined Reason
483 Too Many Hops	Undefined Reason
484 Address Incomplete	Bad Format Address
485 Ambiguous	Undefined Reason
486 Busy Here	In Conference
487 Request Terminated	Undefined Reason
488 Not Acceptable Here	Undefined Reason
500 Internal Server Error	Undefined Reason
501 Not Implemented	Undefined Reason
502 Bad Gateway	Gateway Resource
503 Service Unavailable	Gateway Resource
504 Gateway Timeout	Adaptive Busy
505 Version Not Supported	Invalid Revision
600 Busy Everywhere	Adaptive Busy
603 Decline	Destination Rejection

Q.850 Cause to H.323 Release Complete Reason

SIP Status	H.323 Release Complete Reason
604 Does Not Exist Anywhere	Unreachable Destination
606 Not Acceptable	Undefined Reason

When an call the requires the IWF terminates from the H.323 side and no H.323 Release Complete Reason is specified, the Net-Net SBC maps the Q.850 cause to an H.323 Release Complete Reason using the mapping scheme in the following table. This new H.323 status is then mapped to a SIP status as described in the [IWF Termination from SIP \(446\)](#) table.

Q.850 Cause	H.323 Release Complete Reason
No Route To Destination	Unreachable Destination
Normal Call Clearing	Destination Rejection
User Busy	In Conference
Subscriber Absent	Called Party Not Registered
Invalid Number Format	Bad Format Address
Normal Unspecified	Undefined Reason
No Circuit/Channel Available	No Bandwidth
Network Out Of Order	Unreachable Gatekeeper
Temporary Failure	Adaptive Busy
Switching Equipment Congestion	Gateway Resource
Resource Unavailable	Gatekeeper Resource
Incompatible Destination	Invalid Revision
Interworking Unspecified	No Permission

Codec Mapping

The Net-Net SBC uses the following mapping scheme when converting media specifications between H.245 (used in H.323) and SDP (used in SIP).

Media coming into the Net-Net SBC one way exits the system in the corresponding way as specified in the following table. For example, media coming into the Net-Net SBC as H.245 type g711Ulaw64k exits the system as media type PCMU.

H.245 Type	SDP Media Type
g711Ulaw64k	PCMU
g711Ulaw56k	PCMU
g711Alaw64k	PCMA
g711Alaw56k	PCMA
g726	G726-32
g7231	G723
g722	G722

H.245 Type	SDP Media Type
g728	G728
g729wAnnexB	G729
g729	G729 fmt:18 annexb=no
h261VideoCapability	H261
h263VideoCapability	H263

IWF Service Enhancements

This section describes the Net-Net SBC features that are supported for when the Net-Net SBC performs interworking between SIP and H.323. Enabling these enhancements only requires that you set up a fully functional SIP configuration, a fully functional H.323 configuration, and that you enable IWF on your Net-Net SBC. You do not have to set any special configuration because these enhancements happen automatically.

SIP Redirect—H.323 LRQ Management

When it needs to interact with a SIP Redirect server, the Net-Net SBC can interpret the SIP messages and manage them on the H.323 side of the session. For IWF sessions, the Net-Net SBC handles SIP Redirect and H.323 LRQ messages.



**Redirect—LRQ
Management Sample 1**

This section presents three possible scenarios for SIP Redirect-H.323 LRQ management.

The following diagram shows an established session that uses SIP Redirect—H.323 LRQ management. Here, the Net-Net SBC sends an INVITE to a SIP Redirect Server that responds with a 3xx Redirection message. The Net-Net SBC then sends the gatekeeper/gateway an LCF message that causes an ACF message to be sent to the H.323 endpoint.

**Redirect—LRQ
Management Sample 2**

The following diagram shows how the Net-Net SBC handles the exchange when the SIP Redirect server declares either that there is an error or that there is no such user. These SIP messages come from either the 4xx Request Failure or 5xx Server Failure series. In the example below, the SIP Redirect server returns a 401 Unauthorized message, which the Net-Net SBC interworks and communicates to the H.323 gatekeeper as an LRJ. Then the H.323 gatekeeper/gateway issues an ARJ to the H.323 endpoint.



**Redirect—LRQ
Management Sample 3**

In this call flow, the SIP server issues a 2xx Successful message that is not supposed to be sent (because a 3xx, 4xx, or 5xx message should be sent in response to the Net-Net SBC's INVITE). The Net-Net SBC sends a BYE message to the SIP Redirect Server, but it tries to initiate the session again, this time successfully. The final sample call flow shown rarely occurs.

**SIP INFO and
DTMF UII
Management**

The Net-Net SBC supports DTMF in IWF, enabling features such as keypress, alphanumeric, and hookflash. Because tones are not transmitted as audio, they must pass as out-of-band signaling information, meaning that the Net-Net SBC needs to convert an H.245 UII (User Input Indication) into SIP INFO.

Depending on the capability of the H.323 endpoint, the Net-Net SBC sends either an alphanumeric or DTMF signal in the H.245 UII. The Net-Net SBC sends nothing if the endpoint does not support an alphanumeric or DTMF signal. The SIP INFO message will have a content type of `application/dtmf-relay`, and the message body will be in the form `Signal=*\r\nDuration=250\r\n`. If the duration is absent in the SIP INFO or the UII received on the H.323 side is alphanumeric, the Net-Net SBC uses the a 250 millisecond default value.

Mid-Session Media Change

Mid-session media change happens during a call the requires the IWF when the type of media being sent while a session is in progress changes. For example, a fax transmission might require mid-session media change; besides fax, other applications of this feature are possible. To support the transmission of a T.38 fax sent over an IWF session, some media channels must be opened and others closed. In addition, the Net-Net SBC can accommodate a request for media change from, for example, audio to an image type for T.38 fax.

Because the media requirements are driven by endpoints and Gateways, you do not have to configure the Net-Net SBC's mid-session media change support.

Early Media

For call that require the IWF, the Net-Net SBC supports a cut-through for early media for calls that originate in SIP or H.323.

For a session originating in SIP, the provisional message will contain the SDP information if a Fast Start OLC was received in the Call Proceeding, Alerting, or Progress messages. The same SDP will be sent in the SIP 200 OK.

For a session that starts in H.323, the Net-Net SBC translates the SDP it receives in SIP messages (either a 180 or a 183) into the appropriate H.323 Fast Start elements: Alerting or Progress. If the Alerting or Progress messages contain Fast Start elements, the Progress Indicator Q.931 information element (IE) will also be included in the message with Progress Descriptor 8, indicating that in-band information or an appropriate pattern is now available. This causes the call party to enable end-to-end early media in the reverse direction in accordance with H.323 v4.

In addition, the Net-Net SBC allows early media to flow in the forward direction for a call that requires the IWF starting in H.323 that is being translated to SIP. This happens after the Net-Net SBC has received provisional response with SDP and has sent Alerting or Progress message with Fast Start to the calling party. Similarly, early media in the forward direction is enabled for a call the requires the IWF starting in SIP and being translated to H.323. This happens after the Net-Net SBC received Alerting or Progress messages with Fast Start and maps the Alerting or Progress to SIP 180 or 183 provisional response with the SDP answer.

Display Name Mapping

The Net-Net SBC displays the full name and number of the calling party (for features such as Caller ID) when it handles calls that require the IWF. The Net-Net SBC takes the display name in the From field of the SIP INVITE and maps it to the display IE so that it can show the full name of the calling party.

IWF Ringback Support

When interworking SIP and H.323 to a gateway, PSTN gateway, or other endpoint, the Net-Net SBC uses the mappings shown in the table below. The absence or presence of SDP in the SIP provisional message determines whether the tones are generated in-band or locally.

For each of the mappings listed in the following table, this section provides a sample call flow.

SIP Message	H.323 Message
No Message	CallProceeding
No Message	Progress without PI

SIP Message	H.323 Message
183 with SDP	Progress with PI
180 w/o SDP	Alert without PI
180 with SDP	Alert with PI

In the following diagram, a call the requires the IWF passes through the Net-Net SBC twice, creating two call legs. The call originates from H.323 GW1 and terminates in Phone 1 or Phone 2.



Sample 1: In-band Ringback without Progress Message

This sample flow shows how the Net-Net SBC handles a call that requires the IWF where there is no progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

In this diagram, you can see that the Net-Net SBC maps the progress indicator included in the Alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. When the Progress message appears, it contains the progress indicator rather than the Alerting message containing it.



Sample 2: In-band Ringback with Progress Message

This sample flow shows how the Net-Net SBC handles a call that requires the IWF where there is a progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

For this call flow, you can see again that the Net-Net SBC maps the progress indicator included in the alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. Note that now the Progress message contains the progress indicator.



Sample 3: In-band Ringback without Alerting Message

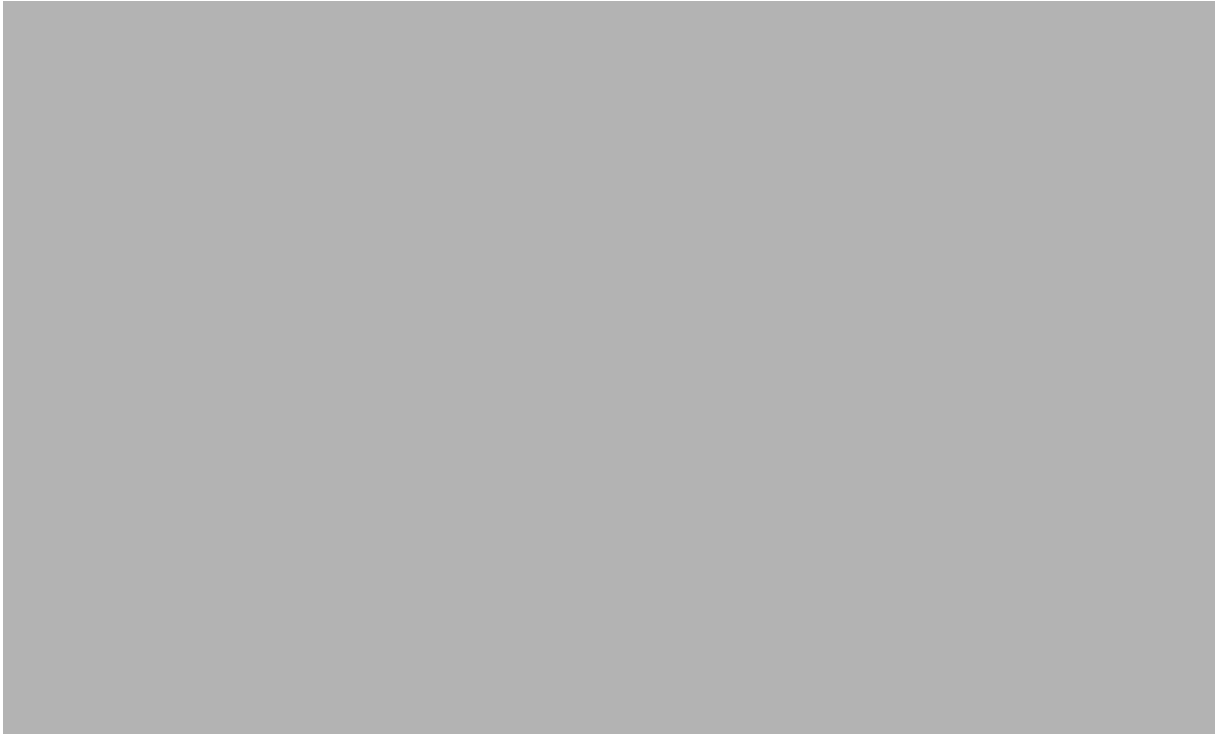
This sample flow shows how the Net-Net SBC handles a call that requires the IWF where there is no progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

In this diagram, you can see that the Net-Net SBC maps the progress indicator included in the Progress message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. When the Alerting message appears, it contains the progress indicator rather than the Progress message containing it.



Sample 4: Out-of-band Ringback without Progress Message

When there is no progress indicator included in the Alerting message, then there is out-of-band ringback. The Net-Net maps the Alerting message to a SIP 180, but it does not include SDP in the SIP 180. This call flow shows that there is no Progress message and that media cannot be set up until after H.323 Connect and SIP messages are sent.



Sample Flow 5: Out-of-band Ringback with Progress Message

When there is no progress indicator included in either the Alerting or Progress messages, then there is out-of-band ringback. The Net-Net maps the Alerting message to a SIP 180, but it does not include SDP in the SIP 180. This call flow shows includes the Progress message; still, media cannot be set up until after H.323 Connect and SIP messages are sent.



H.323 Endpoint- Originated Call Hold and Transfer

When calls that require the IWF originating in H.323, the Net-Net SBC supports call hold, transfer, and conference for the H.323 call leg. The call hold and transfer feature uses signaling procedures based on the ITU-T recommendations/H.323 specification for “third party initiated pause and rerouting.”

You do not have to configure the Net-Net SBC’s call hold and transfer feature.

The following diagram shows how the Net-Net SBC provides call hold and transfer support for IWF sessions that originate in H.323. As you review this section’s call flow diagrams, you might want to refer back to the following logical diagram directly below to review the network elements involved, and what protocols they use.



Basic Call

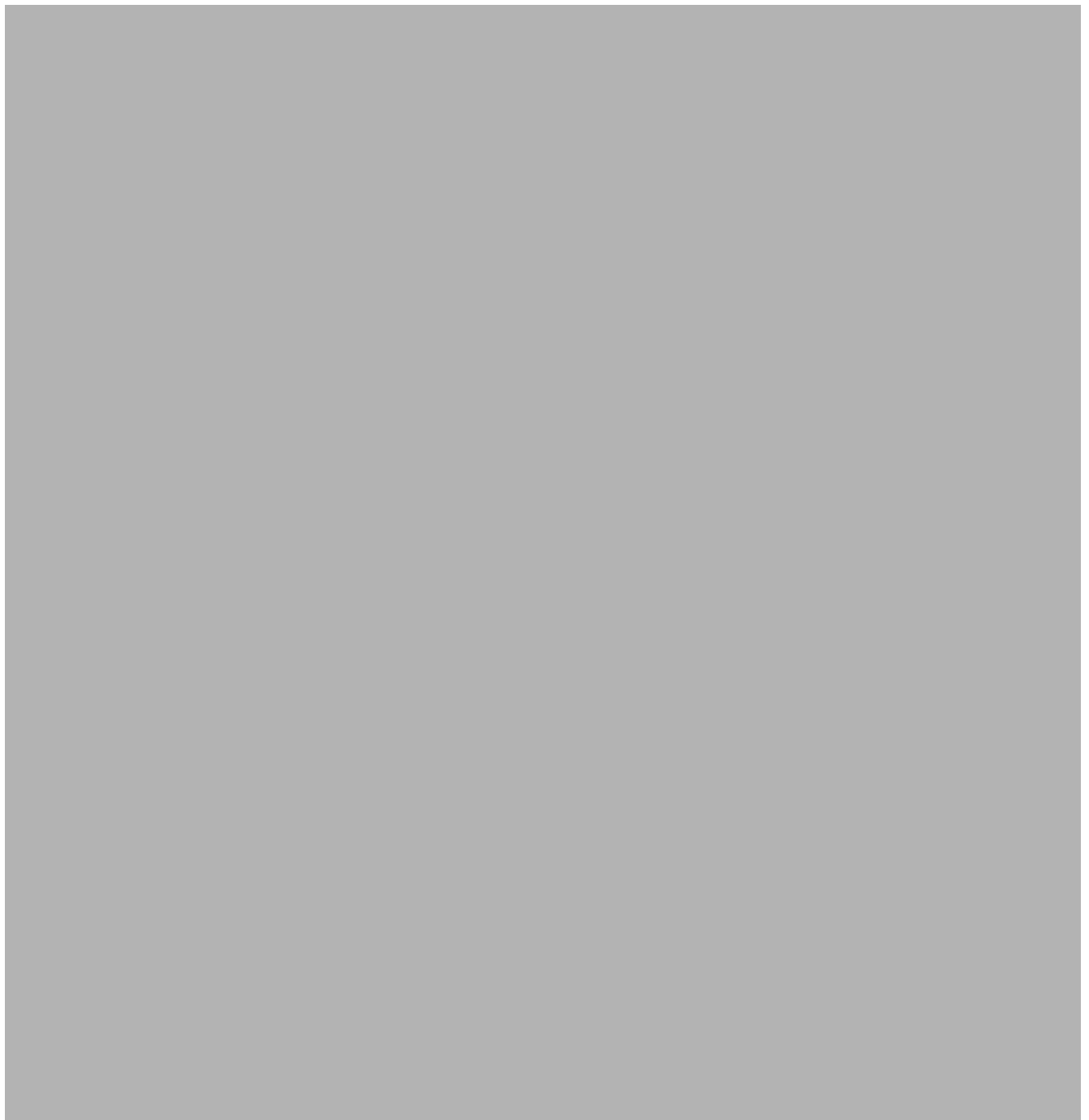
In the following sample basic call, IP PBX A sends an H.323 Slow Starts message ultimately destined for the PSTN through the Net-Net SBC. The Net-Net SBC performs translation to SIP and inserts default information about media. Once the PSTN gateway responds with a 183 containing SDP, the Net-Net SBC sends that information to IP PBX A. Then the Net-Net SBC and the IP PBX exchange TCS- and OLC-related messages, and they negotiate master-slave determination. The Net-Net SBC also sends IP PBX A a Call Progress message with a progress indicator of 8.

After the ringback tone, the proxy sends a 200 OK message with SDP to the Net-Net SBC. The Net-Net SBC sends a Connect message to the IP PBX A, and then it sends another SIP INVITE to the proxy that contains amended SDP (if that information about media is different from the default). After 200 OK and ACK messages are exchanged, media (RTP/RTCP) flow takes place.

**Hold**

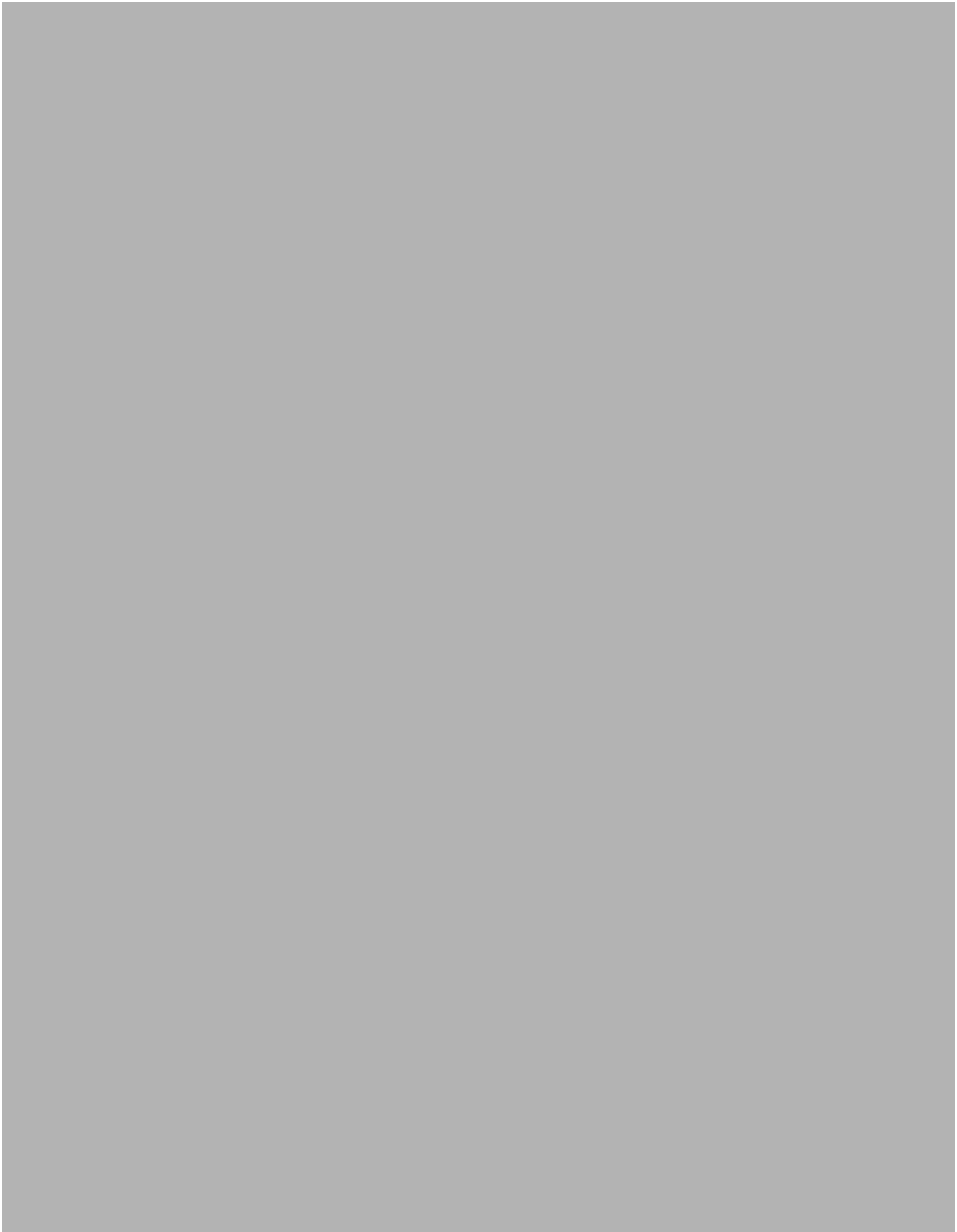
This sample call flow assumes that the IWF call is established and that the RTP/RTCP flow is already in progress. The hold button is pushed, and IP PBX A sends an empty TCS to the Net-Net SBC. The Net-Net SBC puts the called party on hold by sending an INVITE message with 0.0.0.0 SDP to the SIP side of the call. Using 0.0.0.0 as the media address effectively stops the media flow. This INVITE is acknowledged, and the Net-Net SBC closes the channels on the H.323 side, halting the RTP/RTCP flow.

When the caller on the H.323 side takes the call off hold, it resumes with a TCS that the Net-Net SBC receives and then translates on the SIP side as an INVITE with SDP. After that INVITE is acknowledged and received, the Net-Net SBC opens logical channels on the H.323 side and RTP/RTCP flows resume.



Music On Hold

This scenario is similar to the hold feature enabled for calls that require the IWF, except that after the RTP/RTCP flow between the H.323 and SIP sides stops, the call is sent to music on hold. Before the announcement or music plays, the Net-Net SBC sets up the necessary support for media to be exchanged.



Transfer

The call flow described in this section recalls the diagram at the top of the [H.323 Endpoint-Originated Call Hold and Transfer \(458\)](#) section, where endpoints A, B, and C are H.323 devices and endpoint D is a SIP device. When you follow the signaling and media flows, note that there are two Net-Net SBCs in the call transfer and two sets of SIP/H.323 translations that take place. The first Net-Net SBC translates H.323 to SIP, and the second performs the same operations with the protocols reversed.

In the scenario pictured, Party A is on a call with Party D, but wants to transfer Party C to Party D. Party A places Party D on hold, and then makes the call to Party C. Party A then puts Party C on hold, pressing the transfer button. You can see that Net-Net SBC1 receives a TCS from the IP PBX, which is then translated to SIP. Net-Net SBC2 receives it, performs the required protocol translations, and then opens a session with Party C via another IP PBX. Once this session is up and Party D is awakened, channels are established for media exchange.

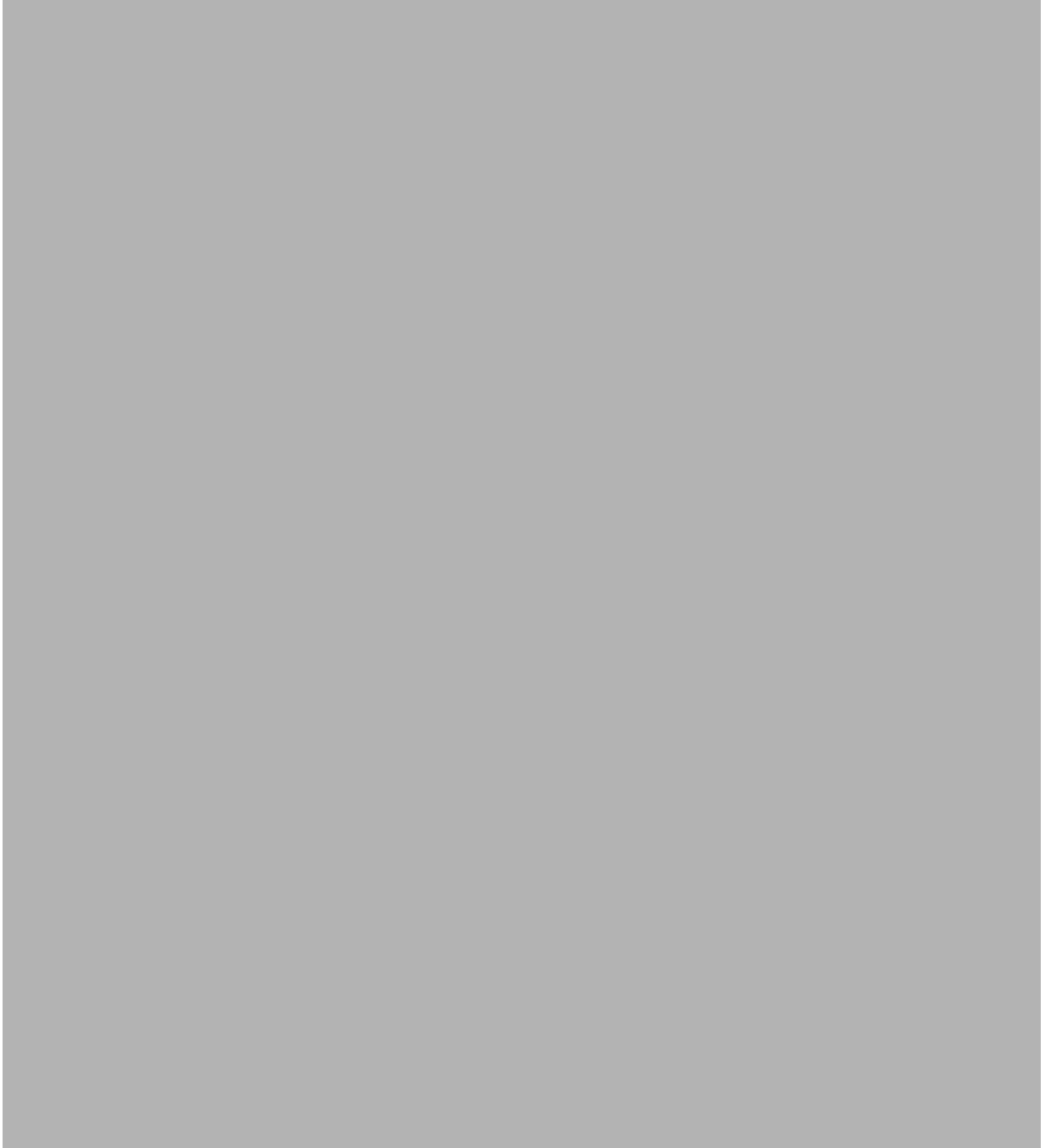
In order to redirect the media so that it flows between Party C and Party D, the Net-Net SBC1 and IP PBX C exchange OLC and OLC Ack messages that contain address information for Party C and for Party D. Address information for both parties is contained in the OLC Ack messages that the Net-Net SBC exchanges with the IP PBX. IP PBX A does not move forward with the call until it has the necessary address information.

Even though Party A's participation in the call stops early in this scenario, the IP PBX with which it is associated keeps the signaling sessions with the Net-Net SBC alive to manage the transfer.



Conference

To conference a call the requires the IWF that starts in H.323, the Net-Net SBC uses a scenario much like the one used for holding a call the requires the IWF. Here again, the INVITE with 0.0.0.0 as the media address and the closing of logical channels stops the flow of RTP/RTCP. After signaling and SDP/media information are re-established, RTP/RTCP for the conference flows.



IWF Call Forwarding

This section describes the Net-Net SBC's IWF Call Forwarding feature, which is supported for calls initiated in SIP that require interworking to H.323.

Prior to the implementation of this feature, the Net-Net SBC did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

- Fail to release the initial call and initiate the forwarded call
- Drop the entire call when the remote endpoint for the call tore down the session

New Behavior

In the diagram below, you can see that the Net-Net SBC sends the initial Setup message to the gateway, and the gateway returns the Facility message with an alternate address for forwarding. Rather than engaging in its former behavior, the Net-Net SBC now releases the call with the gateway and sends a new Setup to the alternate address from the Facility message.

This new Setup up has no effect on the first call leg, which remains connected.



How It Works

When it receives a Facility message with the reason CallForwarded, the Net-Net SBC looks for an alternate transport address in the Facility's alternativeAddress or alternativeAliasAddress element. The Net-Net SBC releases the egress call with the reason facilityCallDeflection. Then it takes one of two courses of action:

- If it does not find an alternative address, the Net-Net SBC releases the ingress call (with 486 BUSY HERE for a call being interworked from SIP to H.323).

If it finds an alternative address and the egress call has not been alerted or answered, the Net-Net SBC at this point tries to initiate a new egress call. The Net-Net SBC uses the alternative alias address to populate the calledPartyNumber information element (IE) and the destination address of the new Setup.

H.323 Sample Call Flow

The following diagram shows how the H.323 Call Forwarding feature works in a purely H.323 environment.

**Media Release for H.323 SS-FS Calls for IWF**

When the Net-Net SBC routes a slow-start to fast-start call, it is possible for the same fast-start call to be routed back through the Net-Net SBC making for a hairpin flow. If it does become a hairpin flow, then the Net-Net SBC routes it to its destination as a fast-start to fast-start call. This can result in one-way media if:

- The destination of the hairpin call is in the same realm as the originating slow-start to fast-start call
- The realm reference in the first bullet item is configured to disable in-realm media management
- The called endpoint accepts the proposed fast-start logical channels

The enhancements to the Net-Net SBC's behavior described in this section show how the Net-Net SBC follows additional procedures when setting up a hairpin flow to avoid one-way media when media release occurs.

How It Works for H.323

For H.323 calls, the Net-Net SBC establishes media using the H.245 procedures described in the H.245 ITU-T recommendation: control protocol for multimedia

communication. It also uses the Fast Connect procedure defined in the H.323 ITU-T recommendation: packet-based multimedia communication systems.

The latter ITU-T recommendation allows a calling endpoint to send a Setup message that contains a fastStart element, a sequence of OLC structures that describe the calling endpoint's proposed forward/reverse logical channels. If the called endpoint accepts this proposal, then logical channels are established.

When the Net-Net SBC translates a call originating in slow-start to fast-start, it uses a Fast Connect procedure in the outgoing leg by sending an outgoing Setup that includes a fastStart element with one or more OLC structures. But when the Net-Net SBC constructs this message, it is unaware of whether the call will become hairpinned or if media release will occur. Because it does not yet have this information, the Net-Net SBC sets the Network Address and the TSAP identifier in the OLC structures to the ingress IP address and port of a corresponding media flow allocated for media traveling between the calling and called endpoints. So if the called endpoint accepts the fastStart the Net-Net SBC proposes, the called endpoint would send its media to the Net-Net SBC. After acceptance, the Net-Net starts H.245 procedures on the slow-start side of the call to set up logical channels on that side. Then the Net-Net SBC updates the IP address and port of the media flows using OLC and OLCAck messages received from the calling endpoint.

This procedure works well for endpoints that are not in the same realm, or that are in the same realm for which media management is disabled, because each endpoint must send its media through the Net-Net SBC. When the endpoints are in the same realm and when media management is enabled, however, the Net-Net SBC must perform additional steps for media release in slow-start to fast-start calls.

To support media release in slow-start to fast-start calls, the Net-Net SBC performs a hold-and-resume procedure on the fast-start side. After it establishes channels on the slow-start side and if it detects media release being enabled, the Net-Net SBC sends an empty TCS to the fast-start side to put that side on hold. Then the called endpoint closes all the logical channels it previously opened in the Fast Connect procedure and stops transmitting to them. And the Net-Net SBC also closes its logical channels. Once the channels are closed, the Net-Net SBC resumes the call by sending a new, restricted TCS to the fast-start side. The restricted TCS only contains the receive and transmit capabilities of the codecs types that the called endpoint accepted in the Fast Connect procedure, and it forces the called endpoint to re-open logical channels of the same codec types accepted in the Fast Connect procedure. Once it receives an OLC from the called endpoint, the Net-Net SBC sends an OLCAck with the Network Address and TSAP identifier for the logical channel from the calling endpoint. Then the Net-Net SBC re-opens logical channels (of the same codec types that it opened in the Fast Connect procedure). If the called endpoint has not changed its Network Address and TSAP identifier for its logical channels, media is re-established after the Net-Net SBC and the called endpoint exit the hold state. The last step is for the Net-Net SBC to re-send the full TCS message from the calling to the called endpoint to inform the called endpoint of the full capabilities of the calling endpoint.

Hold-and-Resume Procedure

The hold-and-resume procedure has three states:

- **Media Hold**—Starts when the Net-Net SBC sends the empty TCS to the called endpoint to put it on hold.

When it detects media release, the Net-Net SBC puts the called endpoint on hold. It can only do so if it has exchanged the TCS/TCSAck messages and completed master-slave determination with the calling endpoint.

When the Net-Net SBC receives a TCSAck in response to the empty TCS that it sent to the called endpoint, it closes the logical channels it opened as part of the Fast Connect procedure; the called endpoint likewise closes its logical channels. The two then exchange CLC and CLCAck messages, which signals the start of the Media Resume state.

- **Media Resume**—Starts when the Net-Net SBC sends a restricted TCS to resume the call.

The restricted TCS the Net-Net SBC sends contains only the receive/transmit capabilities of the codec types previously accepted by the called endpoint in the Fast Connect procedure. This forces the called endpoint to re-open logical channels of the same codec type that were previously accepted in the Fast Connect procedure.

After sending this TCS, the Net-Net is ready (as specified in the ITU-T recommendations) to take part on the master-slave determination (MSD) process. However, the called party and not the Net-Net SBC initiates the MSD if it is required. The MSD is completed if necessary. Alternately, the called endpoint can start to re-open its logical channels. When it receives the first OLC from the called endpoint, the Net-Net SBC also starts to re-open its logical channels.

- **Media Complete**—Starts when all the logical channels that the Net-Net SBC re-opens are acknowledged by the called endpoint.

When it enters the Media Complete state, the Net-Net SBC updates the called endpoint with the full capabilities of the calling endpoint by sending the full TCS.

Additional IWF Steps

For calls originating in slow-start H.323 that require interworking to SIP, the Net-Net SBC also takes addition steps for media release in hairpinned flows that the Net-Net SBC routes as SIP to fast-start H.323.

For such a call, after the Net-Net SBC has established logical channels on the slow-start H.323 side of the call, it sends a reINVITE on the SIP side. This reINVITE has an updated session description to correct the media connection information. The the Net-Net SBC performs the hold-and-resume procedure on the fast-start side of the call. This procedure re-establishes the logical channels between the Net-Net SBC and the called endpoint, avoiding the one-way media problem.

When you are configuring H.323 globally on your Net-Net SBC, you might choose to set the noReInvite option. This option stops the Net-Net SBC from sending a reINVITE after the logical channels are established on the slow-start H.323 side of the call. Instead, the Net-Net SBC's H.323 task communicates internally with its own SIP task a SIP UPDATE message that corrects the SDP; then the SIP task updates media flow destinations. But the Net-Net SBC does not send the UPDATE to the next hop, which can result in the one-way media problem if the call is hairpinned and media release occurs. For such cases, the default behavior for the noReInvite option is overridden. When the Net-Net SBC detects media release in an H.323-SIP call, it forwards the UPDATE to the next hop even when you enable the noReInvite option.

Dependencies

This feature depends on:

- The H.323 endpoint supports the third-party-initiated pause and re-routing feature.
- The H.323 endpoint does not change its Network Address and TSAP identifier when it re-opens the logical channels.
- The H.323 endpoint does not immediately tear down the call when there is not established logical channel in the call.
- The fact that the SIP endpoint supports the UPDATE message if the noReInvite option is enabled.

Before You Configure

The Net-Net SBC's IWF requires that there be complete configurations for both SIP and for H.323. These two sets of configurations function together when the interworking is configured and enabled.

You enable the Net-Net SBC's interworking capability when you set the IWF configuration's state parameter to **enabled**, and all required H.323 and SIP configurations are established. This means that all of the following configurations must be established:

- A full SIP configuration, including SIP interfaces, SIP ports, SIP-NATs (if needed), and SIP features
- A full H.323 configuration, including H.323 global and H.323 interface configurations
- Local policy and local policy attributes (the IWF will not work without these configurations)
- Media profiles
- Session agents and, if needed, session agent groups

H.323 Configuration

You must have a complete configuration to support H.323 traffic on your Net-Net system, including any required support for H.323 Fast Start or Slow Start.

In the H.323 interface configuration, you are able to configure interfaces that enable communication between H.323 devices (for audio, video, and/or data conferencing sessions).

If you know that your Net-Net SBC will be handling traffic originating in Slow Start H.323, you must establish the appropriate list of media profiles in the IWF configuration. Handling Slow Start traffic also requires that you establish appropriate local policy (and local policy attribute) configurations, but configuring session agents and session agent groups is optional.

SIP Configuration

SIP functionality must also be configured on your Net-Net SBC that will perform IWF translations. You must use appropriate local policy (and local policy attribute) configurations, but configuring session agents and session agent groups is optional. If you use session agents, then you must also configure the information you need for media profiles.

For further information about configuring the SIP signaling on your Net-Net SBC, refer to this guide's [SIP Signaling Services \(123\)](#) chapter.

The Role of Local Policy

You must configure local policies (and local policy attributes, if necessary) in order for translations between SIP and H.323 to take place. These local policies determine what protocol is used on the egress side of a session. Local policy and local policy attribute configurations make routing decisions for the session that are based on the next hop parameter that you set. The next hop can be any of the following:

- IPv4 address of a specific endpoint
- Hostname or IPv4 address of a session agent
- Name of a session agent group

You can use the application protocol parameter in the local policy attributes configuration as a way to signal the Net-Net SBC to interwork the protocol of an ingress message into a different protocol as it makes its way to its egress destination (or next hop).

For example, if you set the application protocol parameter to SIP, then an inbound H.323 message will be interworked to SIP as it is sent to the next hop. An inbound SIP message would travel to the next hop unaffected. Likewise, if you set the application protocol parameter to H.323, then an incoming SIP message will be interworked to H.323 before the Net-Net SBC forwards it to the next hop destination.

The following example shows a configured local policy and its attributes used for IWF traffic.

```

local-policy
  from-address          *
  to-address            444
  source-realm          *
  activate-time         N/A
  deactivate-time       N/A
  state                 enabled
  priority              none
  last-modified-date    2007-12-20 17:43:13
policy-attributes
  next-hop              sag:sag_internal
  realm                 internal
  action                none
  terminate-recursion   disabled
  carrier
  start-time            0000
  end-time              2400
  days-of-week          U-S
  cost                  0
  app-protocol          H323
  state                 enabled
  media-profiles

```

Local Policy in an IWF Session Initiated with H.323

In a session where the Net-Net SBC is interworking H.323 to SIP, it internally forwards the session on for interworking when:

- The next hop in the local policy is configured as a SIP session agent
- The next hop in the local policy is configured as a SIP session agent group

- The next hop in the local policy is not configured as a session agent or session agent group, and the application protocol parameter is set to SIP in the local policy attributes configuration.

Local Policy in an IWF Session Initiated with SIP

In a session where the Net-Net SBC is interworking SIP to H.323, it internally forwards the session on for interworking when:

- The next hop in the local policy is configured as an H.323 session agent
- The next hop in the local policy is configured as an H.323 session agent group
- The next hop in the local policy is not configured as a session agent or session agent group, and the application protocol parameter is set to H.323 in the local policy attributes configuration

In this case the local policy should also define the egress realm, which you can set in the realm parameter of the local policy attributes configuration.

Configuring Interworking

If you have already completed the steps outlined in this chapter's [IWF Service Enhancements \(448\)](#) section, then enabling the IWF is a simple process. This section shows you how to enable the IWF, and how to enable certain features that you can use to supplement basic IWF functionality.

An IWF configuration might appear like this in the ACLI:

```
iwf-config
      state                enabled
      add-reason-hdr       disabled
slow-start-no-sdp-in-invite    disabled
      media-profiles
                                PCMU
                                telephone-event
```

ACLI Instructions and Examples

To enable the IWF on your Net-Net SBC:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **iwf-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# iwf-config
```

From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a **?** at the system prompt.
4. **state**—If you want to translate SIP and H.323 sessions on your Net-Net SBC, set this parameter to **enabled**. If you want to turn off the IWF, set this parameter to **disabled** (default).
5. **media-profiles**—Enter the name of the media profiles you want to use for IWF translations. This name is either the name of an SDP codec (such as PCMU), or it can be *telephone-event* if you are configuring your system for DTMF support.

If you want to use more than one media profile for SIP/H.323 translations, enter the names in quotation marks with a space between each one.

```
ACMEPACKET(iwf-config)# media-profiles "PCMU telephone-event"
```

6. **logging**—If you want the Net-Net SBC to log SIP messages that are related to the IWF, set this field to **enabled**. If you do not want to log these messages, leave this field set to **disabled** (default).

IWF Inband Tone Option

This option enables the Net-Net SBC to send a progress indicator (PI) =8 in an H.225 message when an SDP is received in a provisional message. In effect, this option sends network announcements inband. It is also applicable because in some networks H.323 endpoints support early H.245.

The H.323 inband tone option is enabled by adding the **inbandTone** as an option in a configured H.323 stack.

When this option is not used, the ringtone is generated locally (NO PI=8 in PROGRESS OR ALERTING) is the default behavior.

ACLI Instructions and Examples

To configure the IWF inband tone option:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. Use the ACLI **select** command add this feature to an existing H.323 interface.


```
ACMEPACKET(h323-stack)# select
```
6. If you are adding this service to a new H.323 interface, type **options inbandTone** and press <Enter>.


```
ACMEPACKET(h323-stack)# options inbandTone
```
7. If you are adding this service to an H.323 interface that already exists, type **select** to select the interface to which you want to add the service. Then use the options command and prepend the option with a “plus” (+) sign.
 - If you know the name of the interface, you can type the name of the interface at the **name:** prompt and press <Enter>.
 - If you do not know the name of the interface, press <Enter> at the **name:** prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press <Enter>.
 - If are adding service to an existing interface and type in the option without a “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +inbandTone**.

RFC 3326 Support

This section explains the Net-Net SBC's ability to map Q.850 cause values with SIP responses for calls that require IWF.

RFC 3326 defines a header that might be included in any in-dialogue request. This reason header includes cause values that are defined as either a SIP response code or ITU-T Q.850 cause values. You can configure the Net-Net SBC to support sending and receiving RFC 3326 in SIP messages for:

- Mapping H.323 Q.850 cause values to SIP responses with reason header and cause value
- Mapping SIP response messages and RFC 3326 reason header and cause
- Locally generated SIP response with RFC 3326 reason header and cause

As specified in RFC 3326, the Net-Net SBC sends SIP responses to the softswitch that contain the received Q.850 cause code and the reason.

Though the Net-Net SBC can generate RFC 3326 headers, the default behavior for this feature is disabled. Furthermore, the Net-Net SBC can receive and pass SIP error messages (4xx, 5xx, and 6xx) that contain the SIP reason header with a Q.850 cause code and reason (as specified in RFC 3326). If the Net-Net SBC receives an error message without the Reason header, then the Net-Net SBC is not required to insert one.

In calls that require IWF, the Q.850 cause generated in the SIP response are the same as the cause received in the following H.225 messages: Disconnect, Progress, Release, Release Complete, Resume Reject, Status, and Suspend Reject. In addition, the Q.850 cause codes that the Net-Net SBC receives in RFC 3326 headers are passed to the H.323 part of the call unmodified; the H.323 call leg uses this cause code for releasing the call.

For interworking calls between SIP and H.323, you can configure:

- Mappings for SIP status codes to Q.850 values
- Mappings for particular Q.850 cause codes to SIP status codes

If it cannot find the appropriate mapping, then the Net-Net SBC uses default mappings defined in the Default Mappings table below.

The following describes how the Net-Net SBC handles different IWF call scenarios:

- SIP request containing a Reason header—When it receives a request containing a Reason header, the Net-Net SBC determines if the request is a SIP BYE or SIP CANCEL message. RFC 3326 states that the Reason header is mainly used for these types of requests. If there is a Reason header and it contains the Q.850 cause value, then the Net-Net SBC releases the call on the H.323 side using the specified cause value.
- SIP response—When it receives the error response to an initial SIP INVITE, the Net-Net SBC uses its SIP-Q.850 map to determine the Q.850 that it will use to release the call. If there is not a map entry, then the Net-Net SBC uses the default mappings shown in the Default Mappings table.
- Active call released from the H.323 side—If an active call is released from the H.323 side, the Net-Net SBC checks the outgoing realm (the SIP side) to see if the addition of the Reason header is enabled. If it is, then the Net-Net SBC adds the Reason header in the SIP BYE request with the Q.850 value it received from the H.323 side.

- Error during setup of the call on the H.323 side—In the event of an error during setup on the H.323 side of the call, the Net-Net SBC needs to send:
 - An error response, if this is a SIP to H.323 call
 - A SIP CANCEL, if this is a H.323 to SIP call and the H.323 side hangs up before the call is answered on the SIP side

In this case, the Net-Net SBC checks to see if adding the Reason header is enabled in the IWF configuration. If it is, then the Net-Net SBC adds the Reason header with the Q.850 cause value it received from the H.323 side.

- Call released due to a Net-Net SBC error—If the call is released due a Net-Net SBC error and adding the Reason header is enabled in the IWF configuration, the error response to the initial INVITE contains the Reason header. The Net-Net SBC checks the SIP to Q.850 map configurations to determine whether or not the SIP error response code it is generating is configured. If it is, then the Net-Net SBC maps according to the configuration. If it is not, the Net-Net SBC derives cause mapping from the default table.

Like the configuration for SIP-only calls that enable this feature, you can set a parameter in the IWF configuration that enables adding the Reason header in the SIP requests or responses.

Default Mappings

This table defines the default mappings the Net-Net SBC uses when it cannot locate an appropriate entry that you have configured.

Q.850 Cause Value		SIP Status		Comments
1	Unallocated number	404	Not found	
2	No route to specified transit network	404	Not found	
3	No route destination	404	Not found	
16	Normal calling clearing		BYE message	A call clearing BYE message containing cause value 16 normally results in the sending of a SIP BYE or CANCEL request. However, if a SIP response is to be sent to the INVITE request, the default response code should be used.
17	User busy	486	Busy here	
18	No user responding	408	Request timeout	
19	No answer from the user	480	Temporarily unavailable	
20	Subscriber absent	480	Temporarily unavailable	

Q.850 Cause Value		SIP Status		Comments
21	Call rejected	603	Decline (if location filed in Cause information element indicates user; otherwise 403 Forbidden is used)	
22	Number changed	301	Moved permanently (if information in diagnostic field of Cause information element is suitable for generating SIP Contact header; otherwise 410 Gone is used)	
23	Redirection to new destination	410	Gone	
25	Exchange routing error	483	Too many hops	
27	Destination out of order	502	Bad gateway	
28	Address incomplete	484	Address incomplete	
29	Facility rejected	501	Not implemented	
31	Normal, unspecified	480	Temporarily unavailable	
34	No circuit, channel unavailable	503	Service unavailable	
38	Network out of order	503	Service unavailable	
41	Temporary failure	503	Service unavailable	
42	Switching equipment congestion	503	Service unavailable	
47	Resource unavailable unspecified	503	Service unavailable	
55	Incoming calls barred with CUG	403	Forbidden	
57	Bearer capability not authorized	403	Forbidden	
58	Bearer capability not presently available	503	Service unavailable	
65	Bearer capability not implemented	488	Not acceptable here	
69	Requested facility not implemented	501	Not implemented	

Q.850 Cause Value		SIP Status		Comments
70	Only restricted digital information available	488	Not acceptable here	
79	Service or option not implemented, unspecified	501	Not implemented	
87	User not member of CUG	403	Forbidden	
88	Incompatible destination	503	Service unavailable	
102	Recovery on timer expiry	504	Server time-out	

ACLI Instructions and Examples

To configure a SIP status to Q.850 Reason with cause mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-q850-map** and press <Enter>.


```
ACMEPACKET(session-router)# sip-q850-map
ACMEPACKET(sip-q850-map)#
```
4. Type **entries** and press <Enter>.


```
ACMEPACKET(sip-q850-map)# entries
ACMEPACKET(sip-q850-map-entry)#
```

From here, you can view the entire menu for the SIP status to Q.850 Reason with cause mapping entries configuration by typing a **?**.
5. **sip-status**—Set the SIP response code that you want to map to a particular Q.850 cause code and reason. There is no default, and the valid range for values is 100-699.
6. **q850-cause**—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
7. **q850-reason**—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

To configure a Q.850 cause to a SIP status with reason mapping:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-q850-map** and press <Enter>.


```
ACMEPACKET(session-router)# q850-sip-map
ACMEPACKET(q850-sip-map)#
```
4. Type **entries** and press <Enter>.


```
ACMEPACKET(q850-sip-map)# entries
ACMEPACKET(q850-sip-map-entry)#
```

From here, you can view the entire menu for the Q.850 cause to a SIP response code with reason mapping entries configuration by typing a ?.
5. **q850-cause**—Set the Q.850 cause code that you want to map to a SIP status with reason. There is no default.
6. **sip-status**—Set the SIP response code to which you want to map the Q.850 cause that you set in step 5. There is no default, and the valid range for a value is 100-699.
7. **sip-reason**—Set the reason that you want to use with the SIP response code that you specified in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.

To enable the Net-Net SBC to add the Reason header for calls that require IWF:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **iwf-config** and press <Enter>.


```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```
4. **add-reason-header**—Set this parameter to **enabled** to turn add the Reason header. The default is **disabled**.

IWF Privacy: Caller Privacy on Unsecure Networks

This feature enables bi-directional SIP/H.323 IWF support for CPID hiding by using the presentation indicators in the Calling Party Number information element for H.323 signaling, and RFC 3325-based privacy support for SIP signaling. It lets the Net-Net SBC insert the P-Asserted-Identity and the Privacy header in the INVITE when the presentation indicator is set to **restricted**.

The presence, or absence, of P-Asserted-Identity and Privacy headers in the SIP INVITE informs the remote SIP proxy or endpoint to either block or advertise the CPID.

About the Presentation Indicator

When address information represents a telephone number, the relevant information can appear in the Calling Party Number information element (IE). This IE contains the caller's number, information about the number, and presentation and screening indicators found in octet 3a. In order to prevent a calling party number to be passed through, the presentation indicator parameter (octet 3a) in the Calling Party IE must be set to a value other than 00.

In a H.323 to SIP IWF call, octet 3a in the Q.931 message indicates the caller's preference for CPID restriction. If bits 7 and 6 are set to (0 1), the presentation is restricted and the outbound SIP INVITE from the IWF stack must be constructed as such.

H.323 to SIP IWF Call

When the presentation indicator in the calling party IE is set to restricted, the INVITE's From and Contact headers sent from to sipd will be modified according to RFC 3325. When the Net-Net SBC receives calls initiated as H.323, it will recognize the caller's presentation bits as defined in Q.931 and use that information to construct a SIP INVITE in accordance with the user's indicated preference.

- Inclusion of a P-Asserted-Identity header in the INVITE, containing the calling party's CPID and the Net-Net SBC's IP address, constructed as a SIP URI (same mechanism used to construct the From-URI today).
- Addition of a Privacy header with its value set to "id". This addition indicates to the upstream proxies and gateways that the caller address is to be hidden.

The sipd will either proxy or strip these headers according to RFC 3325, depending on the SIP interface and SIP session agent configurations.

Example 1: SETUP Sent from h323d to Remote H.323 Endpoints

```
Q.931
Protocol discriminator: Q.931
Call reference value length: 2
Call reference flag: Message sent from originating side
Call reference value: 2F62
Message type: SETUP (0x05)
Bearer capability
Information element: Bearer capability
Length: 3
...0 1000 = Information transfer capability: Unrestricted digital
information (0x08)
.00. .... = Coding standard: ITU-T standardized coding (0x00)
1... .... = Extension indicator: last octet
...1 0011 = Information transfer rate: 384 kbit/s (0x13)
.00. .... = Transfer mode: Circuit mode (0x00)
1... .... = Extension indicator: last octet
...0 0101 = User information layer 1 protocol: Recommendation H.221 and
H.242 (0x05)
1... .... = Extension indicator: last octet
Display 'jdoe\000'
Information element: Display
Length: 9
Display information: jdoe\000
```

Calling party number
 Information element: Calling party number
 Length: 2
 0000 = Numbering plan: Unknown (0x00)
 .000 = Number type: Unknown (0x00)
 0... = Extension indicator: information continues through the next
 octet
00 = Screening indicator: User-provided, not screened (0x00)
.01. = Presentation indicator: Presentation restricted (0x01)
 1... = Extension indicator: last octet

Example 2: INVITE from h323d to sipd

The two new headers will be stripped by the sipd when the INVITE is sent to a untrusted SIP proxy or endpoint and will be proxied over to a trusted SIP proxy or end point.

```
INVITE sip:780@192.168.200.6:5060;acme_realm=internal SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5070;branch=z9hG4bKIWF00000510d031s9kou5c0;acme_irealm=external
Contact: "Anonymous"<sip:anonymous@127.0.0.1:5070
GenericID: 74000000@000825010100
Supported: 100rel
From: "Anonymous"<sip:anonymous@anonymous.invalid>;tag=0000004a000d8cc0
To: <sip:780@192.168.200.6:5060
Call-ID: 7f00000113ce0000004a000d88d8@127.0.0.1
CSeq: 2 INVITE
P-Asserted-Identity: "jdoe"<sip:42343@192.168.200.68:5060>
Privacy: id
Content-Length: 175
Content-Type: application/sdp

v=0
o=IWF 3 3 IN IP4 192.168.1.6
s=H323 Call
c=IN IP4 192.168.1.6
t=0 0
m=audio 5666 RTP/AVP 0 101 18
a=rtpmap:0 PCMU/8000/1
a=rtpmap:101 telephone-event/8000/1
a=fmtp:101 0-15
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
m=video 5668 RTP/AVP 31
a=rtpmap:31 H261/9000/1
```

SIP to H.323

For a SIP to H.323 call, the Net-Net SBC must recognize the caller's Privacy request and set the presentation bits accordingly when constructing the outbound RAS/SETUP message. It must check SIP calls for the Privacy header (with value set

to “id”). If this header is present, the SETUP’s octet 3a’s presentation bits must be set to restricted.

The Net-Net SBC does not support any other value for the Privacy header. For those calls, the SETUP will not include a presentation indicator.

Example: INVITE from SIP End Point to sipd

Apr 21 08:50:38.786 On [0:0]192.168.200.68:5060 received from 192.168.200.6:5062

INVITE sip:800@192.168.200.68:5060 SIP/2.0

Via: SIP/2.0/UDP 192.168.200.6:5062

From: anonymous <sip:anonymous@192.168.200.6:5062>;tag=1

To: sut <sip:800@192.168.200.68:5060

P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062

Privacy: id

Call-ID: 1.1688.192.168.200.6@sipp.call.id

Cseq: 1 INVITE

Contact: sip:anonymous@192.168.200.6:5062

Max-Forwards: 70

Subject: Performance Test

Content-Type: application/sdp

Content-Length: 136

v=0

o=user1 53655765 2353687637 IN IP4 127.0.0.1

s=-

t=0 0

c=IN IP4 127.0.0.1

m=audio 10000 RTP/AVP 0

a=rtpmap:0 PCMU/8000

Sample INVITE from sipd to h323d

Apr 21 08:50:38.807 On 127.0.0.1:5070 received from 127.0.0.1:5060

INVITE sip:800@127.0.0.1:5070;acme_sag=sag1;acme_irealm=internal SIP/2.0

Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK0804o700c0f0t9gpj0g0.1

From: anonymous <sip:anonymous@192.168.200.6:5062>;tag=SDm8kvc01-1

To: sut <sip:800@192.168.200.68:5060

P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062

Privacy: id

Call-ID: SDm8kvc01-083221d8c0fa33f71ae85dd6ed0e4ea4-06ahc21

Cseq: 1 INVITE

Contact: <sip:anonymous@192.168.200.68:5060;transport=udp

Max-Forwards: 69

Subject: Performance Test

Content-Type: application/sdp

Content-Length: 136

GenericID: 9883100005@000825010100


```

v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4      127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Sample SETUP sent from h323d to remote H323 EP
Q.931
  Protocol discriminator: Q.931
  Call reference value length: 2
  Call reference flag: Message sent from originating side
  Call reference value: 664D
  Message type: SETUP (0x05)
  Bearer capability
    Information element: Bearer capability
    Length: 3
    ...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)
    .00. .... = Coding standard: ITU-T standardized coding (0x00)
    1... .... = Extension indicator: last octet
    ...1 0000 = Information transfer rate: 64 kbit/s (0x10)
    .00. .... = Transfer mode: Circuit mode (0x00)
    1... .... = Extension indicator: last octet
    ...0 0011 = User information layer 1 protocol: Recommendation G.711
  A-law (0x03)
  1... .... = Extension indicator: last octet
  Display 'anonymous'
    Information element: Display
    Length: 9
    Display information: anonymous
  Calling party number
    Information element: Calling party number
  Length: 2
    .... 0000 = Numbering plan: Unknown (0x00)
    .000 .... = Number type: Unknown (0x00)
    0... .... = Extension indicator: information continues through the
  next octet
    .... ..00 = Screening indicator: User-provided, not screened (0x00)
    .01. .... = Presentation indicator: Presentation restricted (0x01)
    1... .... = Extension indicator: last octet

```

IWF Privacy: Caller Privacy on Secure Connections

In prior releases, when the H.323 endpoint sends a SETUP with presentation indicator set to `allowed`, the Net-Net SBC does not insert the P-Asserted-Identity in the INVITE. The SIP INVITE needs the P-Asserted-Identity header to support calling line identification presentation (CLIP) to calling line identification restriction (CLIR) in an IP multimedia subsystem (IMS) solution. This feature lets the Net-Net

SBC insert the P-Asserted-Identity in the INVITE when the presentation indicator is set to **allowed**.

- CLIP is a service provided to the called party that allows the display of the calling number (caller ID). The user-provided calling number must be transported from the caller to the called party.
- CLIR is a service provided to the calling party that lets it indicate whether or not the calling number is to be displayed to the called party. It sets a calling number presentation indicator to **allowed** or **restricted**. Regulations require that network administrations remove the calling number before it is sent to the called party, if the calling party has so requested.

H.323 to SIP IWF

When the Net-Net SBC translates incoming H.323 messages to SIP on a secure connection (which means the Net-Net SBC can rely on the data sent from the originator); it will translate the information in the H.323 messages into SIP messages as detailed in the following sections.

Calls with Presentation Allowed

When the Net-Net SBC receives a SETUP from the H.323 domain where presentation is allowed, it generates an INVITE to the SIP domain with the following header. (Presentation is allowed when the calling party's information element presentation indicator (octet 3a) equals **00**.)

- P-Asserted-ID: the userpart should be derived from the Calling Party Number Information Element digits.

H.323 to SIP

When h323d receives a SETUP with the calling party's information element presentation indicator set to **allowed**, the Net-Net SBC will add the P-Asserted-Identity header to the INVITE. The P-Asserted-Identity is very similar to the FROM header, except for the tag.

Sample SETUP sent from h323d to Remote H323 Endpoints

```

Q.931
Protocol discriminator: Q.931
Call reference value length: 2
Call reference flag: Message sent from originating side
Call reference value: 2F62
Message type: SETUP (0x05)
Bearer capability
Information element: Bearer capability
Length: 3
...0 1000 = Information transfer capability: Unrestricted digital
information (0x08)
.00. .... = Coding standard: ITU-T standardized coding (0x00)
1... .... = Extension indicator: last octet
...1 0011 = Information transfer rate: 384 kbit/s (0x13)
.00. .... = Transfer mode: Circuit mode (0x00)
1... .... = Extension indicator: last octet
...0 0101 = User information layer 1 protocol: Recommendation H.221 and
H.242 (0x05)
1... .... = Extension indicator: last octet
Display 'jdoe\000'
Information element: Display
Length: 9
Display information: jdoe\000
Calling party number: '42343'
Information element: Calling party number
Length: 6
.... 1001 = Numbering plan: Private numbering (0x09)
.110 .... = Number type: Abbreviated number (0x06)
0... .... = Extension indicator: information continues through the next
octet
.... ..00 = Screening indicator: User-provided, not screened (0x00)
.00. .... = Presentation indicator: Presentation allowed (0x00)
1... .... = Extension indicator: last octet
Calling party number digits: 42343

```

SIP to H.323

When the sipd receives an INVITE with the P-Asserted-Identity header but without the Privacy header, the Net-Net SBC will set the presentation indicator to **allowed** in H.323's SETUP.

When the Privacy header is present with the value "id", the presentation indicator will be set to **restricted**. The Net-Net SBC does not support any other value for the Privacy header and so for those call flows, the presentation indicator will be absent in the SETUP.

Example 1: INVITE from sip EP to sipd

```

Apr 20 04:43:54.220 On [0:0]192.168.200.68:5060 received from
192.168.200.6:5062
INVITE sip:800@192.168.200.68:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.6:5062
From: sipp <sip:7789@192.168.200.6:5062>;tag=1
To: sut <sip:800@192.168.200.68:5060>
P-Asserted-Identity: sipp <sip:7789@192.168.200.6:5062>
Call-ID: 1.1336.192.168.200.6@sipp.call.id
Cseq: 1 INVITE
Contact: sip:7789@192.168.200.6:5062
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
^M
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4 127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

```

Example: INVITE from sipd to h323d

```

Apr 20 04:43:54.240 On 127.0.0.1:5070 received from 127.0.0.1:5060
INVITE sip:800@127.0.0.1:5070;acme_sag=sag1;acme_irealm=internal SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK000c0210385hv9gpt001.1
From: sipp <sip:7789@192.168.200.6:5062>;tag=SDk0jpc01-1
To: sut <sip:800@192.168.200.68:5060>
Call-ID: SDk0jpc01-8e15e11e7f9a20523462972843c7e579-06ahc21
Cseq: 1 INVITE
Contact: <sip:7789@192.168.200.68:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 136
GenericID: 160400004@000825010100

v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4 127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

```

Sample SETUP sent from h323d to remote H323 EP

Q.931

Protocol discriminator: Q.931

Call reference value length: 2

Call reference flag: Message sent from originating side

Call reference value: 664D

Message type: SETUP (0x05)

Bearer capability

Information element: Bearer capability

Length: 3

...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)

.00. = Coding standard: ITU-T standardized coding (0x00)

1... = Extension indicator: last octet

...1 0000 = Information transfer rate: 64 kbit/s (0x10)

.00. = Transfer mode: Circuit mode (0x00)

1... = Extension indicator: last octet

...0 0011 = User information layer 1 protocol: Recommendation G.711

A-law (0x03)

1... = Extension indicator: last octet

Display 'sipp'

Information element: Display

Length: 4

Display information: sipp

Calling party number: '7789'

Information element: Calling party number

Length: 6

.... 1001 = Numbering plan: Private numbering (0x09)

.110 = Number type: Abbreviated number (0x06)

0... = Extension indicator: information continues through the next octet

.... ..00 = Screening indicator: User-provided, not screened (0x00)

.00. = **Presentation indicator: Presentation all** 1... = Extension indicator: last octet

Calling party number digits: 7789

IWF Privacy Extensions for Asserted Identity in Untrusted Networks

For IWF privacy, the Net-Net SBC supports:

- IWF caller privacy on unsecure networks—A variant of RFC 3325, where the P-Asserted-Id is inserted when the presentation indicator is set to allowed. This feature enables bi-directional SIP/H.323 IWF support for CPID hiding by using the presentation indicators in the Calling Party Number information element for H.323 signaling, and RFC 3325-based privacy support for SIP signaling. It allows the Net-Net SBC to insert the P-Asserted-Identity and the Privacy header in the INVITE when the presentation indicator is set to restricted.

The presence, or absence, of P-Asserted-Identity and Privacy headers in the SIP INVITE informs the remote SIP proxy or endpoint to either block or advertise the CPID.

- IWF caller privacy on secure connections—When the H.323 endpoint sends a SETUP with presentation indicator set to allowed, the Net-Net SBC does not insert the P-Asserted-Identity in the INVITE. The SIP INVITE needs the P-

Asserted-Identity header to support calling line identification presentation (CLIP) to calling line identification restriction (CLIR) in an IP multimedia subsystem (IMS) solution. This feature allows the Net-Net SBC to insert the P-Asserted-Identity in the INVITE when the presentation indicator is set to allowed.

Now the Net-Net SBC supports an enhancement to IWF caller privacy where the P-Preferred-Identity is inserted instead of the P-Asserted-Identity.

In this implementation, when the incoming H.323 Setup message has a presentation indicator set to restricted and the ingress H.323 session agent has the new PPreferredId option configured, the Net-Net SBC sends the Privacy header with P-Preferred-Identity (instead of P-Asserted-Identity).

IWF Call Originating in H.323

This section shows an example H.323 Setup that arrives from an H.323 endpoint, and how the Net-Net SBC adds the P-Preferred-Identity header (which has calling party number information) and the Privacy header to the SIP INVITE.

Sample H.323 Setup from a Remote Endpoint

```

Q.931
  Protocol discriminator: Q.931
  Call reference value length: 2
  Call reference flag: Message sent from originating side
  Call reference value: 2FB6
  Message type: SETUP (0x05)
  Bearer capability
    Information element: Bearer capability
    Length: 3
    ...0 1000 = Information transfer capability: Unrestricted digital information
(0x08)
    .00. .... = Coding standard: ITU-T standardized coding (0x00)
    1... .... = Extension indicator: last octet
    ...1 0011 = Information transfer rate: 384 kbit/s (0x13)
    .00. .... = Transfer mode: Circuit mode (0x00)
    1... .... = Extension indicator: last octet
    ...0 0101 = User information layer 1 protocol: Recommendation H.221 and H.242 (0x05)
    1... .... = Extension indicator: last octet
  Display 'rdoe\000'
    Information element: Display
    Length: 9
    Display information: jdoe\000
  Calling party number: '42343'
    Information element: Calling party number
    Length: 6
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .000 .... = Number type: Unknown (0x00)
    0... .... = Extension indicator: information continues through the next octet
    .... ..00 = Screening indicator: User-provided, not screened (0x00)
    .01. .... = Presentation indicator: Presentation restricted (0x01)
    1... .... = Extension indicator: last octet
    Calling party number digits: 42343
  E.164 Calling party number digits: 42343
  Called party number: '780'
    Information element: Called party number
    Length: 4
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .000 .... = Number type: Unknown (0x00)
    1... .... = Extension indicator: last octet
    Called party number digits: 780
  E.164 Called party number digits: 780
  User-user
    Information element: User-user
  
```

Length: 161
 Protocol discriminator: X.208 and X.209 coded user information

Sample SIP INVITE from the Net-Net SBC to a SIP Endpoint

```
Aug 29 15:46:25.214 On [0:0]192.168.200.68:5060 sent to
192.168.200.6:5060
INVITE sip:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.68:5060;branch=z9hG4bK6810pr20205h2akqe381.1
Contact: "Anonymous"<sip:anonymous@192.168.200.68:5060;transport=udp>
Supported: 100rel
From: "Anonymous"<sip:anonymous@anonymous.invalid>;tag=SDfd9sa01-
000000ba00023280
To: <sip:780@192.168.200.6:5060>
Call-ID: SDfd9sa01-6f93292521b83a0980647f34451c5afd-06ahc21
CSeq: 2 INVITE
P-Preferred-Identity: "rdoe"<sip:42343@192.168.200.68:5060>
<b>Privacy: id<\b>
Content-Length: 180
Content-Type: application/sdp
Max-Forwards: 70

v=0
o=IWF 5 5 IN IP4 192.168.200.5
s=H323 Call
c=IN IP4 192.168.200.65
t=0 0
m=audio 5010 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1
m=video 5014 RTP/AVP 31
a=rtpmap:31 H261/9000/1
```

Before You Configure

Before you configure your Net-Net SBC to support this feature, note the following considerations:

- The ingress H.323 session agent cannot be configured with the `NoPAssertedId` option
- For use in Release 4.1.1 and higher, the global SIP configuration should be configured with the `disable-ppi-to-pai` option; the older `disable-privacy` option will also work

ACLI Instructions and Examples

To enable the inclusion of P-Preferred-Identity:

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


```
ACMEPACKET# configure terminal
```
2. Type `session-router` and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type `session-agent` and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Select the session agent where you want to apply this feature.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: 204.12.60.5      realm=private
2: 124.21.5.3      realm=public
```

```
selection:1
ACMEPACKET(session-agent)#
```

5. **options**—Set the options parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**PPreferredId**), and then press <Enter>.

```
ACMEPACKET(realm-config)# options +PPreferredId
```

If you type **options PPreferredId**, you will overwrite any previously configured options. In order to append the new option to the session agent's options list, you must prepend the new option with a "plus" sign as shown in the previous example.

6. Save and activate your configuration.

IWF Privacy for Business Trunking

The Net-Net SBC supports [IWF Privacy: Caller Privacy on Unsecure Networks \(477\)](#) and [IWF Privacy: Caller Privacy on Secure Connections \(481\)](#), but IWF Privacy for Business Trunking, supports the case where SIP and H.323 PBXs are connected to the core IMS system. Traffic originated at the IP PBXs terminates either at other PBXs or at the PSTN, and includes the possibility of accepting incoming traffic from the PSTN. CLIP and CLIR must be supported for calls in either direction for calls that require interworking between SIP and H.323. Unlike the two features described above, this new feature supports the fact that only a network-based application server has sufficient privilege to assert the identity of the calling party.

Thus, for this feature, the Net-Net SBC does not force privacy. Instead, the implemented feature assumes that the H.323 session agent is an IP PBX, and the Net-Net SBC only indicates to the SIP core that privacy is being requested. In other words, the Net-Net SBC is not required to interwork the H.323 presentation indicator parameter to RFC 3325 by including the P-Asserted-Identity header. The indication to the SIP core that privacy is being requested excludes identity assertion.

You configure this feature using two session agent options:

- **allowCPN**—Set in the egress H.323 session agent, allows the Net-Net SBC to send the calling party number information element (IE), even when the presentation indicator is set to **restricted**.
- **NoPAssertedId**—Set in the ingress H.323 session agent; when the incoming SETUP message has the presentation indicator is set to **restricted**, instructs the Net-Net SBC to send a Privacy header without the P-Asserted-Identity and not to make the From header anonymous.

A Call Originating in H.323

This section describes for the IWF Privacy for Business trunking feature works for a call originating in H.323 that requires interworking to SIP.

When the Net-Net SBC receives an H.323 SETUP with a presentation indicator of the calling party information element (IE) is set to **restricted** and this SETUP was received from a session agent is configured with the **NoPAssertedID** option, the Net-Net SBC only adds the Privacy header with the value **no**. In this case, there will be no P-Asserted-Identity and the From header will contain the calling Party

information that was extracted from the callingPartyIE. The Net-Net SBC assumes that the PBX will send the callingPartyNumber in the IE, even though it would like to have the calling party number restricted.

Sample SETUP Message from an H.323 Endpoint

```

Q.931
  Protocol discriminator: Q.931
  Call reference value length: 2
  Call reference flag: Message sent from originating side
  Call reference value: 2FB6
  Message type: SETUP (0x05)
  Bearer capability
    Information element: Bearer capability
  Length: 3
    ...0 1000 = Information transfer capability: Unrestricted digital
    information (0x08)
    .00. .... = Coding standard: ITU-T standardized coding (0x00)
    1... .... = Extension indicator: last octet
    ...1 0011 = Information transfer rate: 384 kbit/s (0x13)
    .00. .... = Transfer mode: Circuit mode (0x00)
    1... .... = Extension indicator: last octet
    ...0 0101 = User information layer 1 protocol: Recommendation
    H.221 and H.242 (0x05)
    1... .... = Extension indicator: last octet
    Display 'jdoe\000'
    Information element: Display
  Length: 9
    Display information: jdoe\000
    Calling party number: '42343'
    Information element: Calling party number
    Length: 6
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .000 .... = Number type: Unknown (0x00)
    0... .... = Extension indicator: information continues through
    the next octet
    .... ..00 = Screening indicator: User-provided, not screened
    (0x00)
    .01. .... = Presentation indicator: Presentation restricted
    (0x01)
    1... .... = Extension indicator: last octet
    Calling party number digits: 42343
    E.164 Calling party number digits: 42343
    Called party number: '780'
    Information element: Called party number
    Length: 4
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .000 .... = Number type: Unknown (0x00)
    1... .... = Extension indicator: last octet
    Called party number digits: 780
    E.164 Called party number digits: 780
  User-user
    Information element: User-user
    Length: 161
    Protocol discriminator: X.208 and X.209 coded user information
  
```

Sample INVITE from the Net-Net SBC to the SIP Endpoint

```

May  5 15:11:51.996 On [0:0]192.168.200.68:5060 sent to
192.168.200.6:5060
INVITE sip:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.68:5060;branch=z9hG4bK00020a20eg11s94pg700.1
Contact: "jdoe"<sip:42343@192.168.200.68:5060>;transport=udp>
Supported: 100rel
From: "jdoe"<sip:42343@192.168.200.68:5060>;tag=SDetur801-
00000194000e2ce8
To: <sip:780@192.168.200.6:5060>
Call-ID: SDetur801-231c7b30909ca525ce12cbfeb57754ea-06ahc21
CSeq: 2 INVITE
Privacy: id
Content-Length: 231
Content-Type: application/sdp
Max-Forwards: 70

v=0
o=IWF 2 2 IN IP4 192.168.200.65
s=H323 Call
c=IN IP4 192.168.200.65
t=0 0
m=audio 5004 RTP/AVP 8 0
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000/1
m=video 5006 RTP/AVP 31 34
a=rtpmap:31 H261/8000
a=rtpmap:34 H263/9000/1

```

A Call Originating in SIP

This section describes for the IWF Privacy for Business trunking feature works for a call originating in SIP that requires interworking to H.323.

When the Net-Net SBC receives a SIP INVITE with a Privacy header that has the value **id**, it sets the presentation indicator to **restricted** in the corresponding H.323 SETUP message. If the H.323 session agent is configured with the **allowCPN** option, the Net-Net SBC sends the display IE and the calling party number to the H.323 session agent. If that option is not set in the H.323 session agent, then the Net-Net SBC reverts to its default behavior, which is to not to send the display IE and to hide the calling party number.

Sample INVITE from a SIP Endpoint to the Net-Net SBC

```

May  5 14:41:54.513 On [0:0]192.168.200.68:5060 received from
192.168.200.6:5060
INVITE sip:800@192.168.200.68:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.6:5060
From: sipp <sip:sipp@192.168.200.6:5060>;tag=1
To: sut <sip:800@192.168.200.68:5060>
Call-ID: 1.3068.192.168.200.6@sipp.call.id
Cseq: 1 INVITE
Contact: sip:sipp@192.168.200.6:5060
Privacy: id
P-Asserted-Identity: sipp <sip:1234@192.168.200.6:5060>
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp

```

Content-Length: 136

```
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
t=0 0
c=IN IP4 127.0.0.1
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Sample SETUP from the Net-Net SBC to the H.323 Endpoint

```
Q.931
  Protocol discriminator: Q.931
  Call reference value length: 2
  Call reference flag: Message sent from originating side
  Call reference value: 44B0
  Message type: SETUP (0x05)
  Bearer capability
    Information element: Bearer capability
    Length: 3
    ...1 0000 = Information transfer capability: 3.1 kHz audio (0x10)
    .00. .... = Coding standard: ITU-T standardized coding (0x00)
    1... .... = Extension indicator: last octet
    ...1 0000 = Information transfer rate: 64 kbit/s (0x10)
    .00. .... = Transfer mode: Circuit mode (0x00)
    1... .... = Extension indicator: last octet
    ...0 0011 = User information layer 1 protocol: Recommendation
G.711 A-law (0x03)
    1... .... = Extension indicator: last octet
  Display 'sipp'
    Information element: Display
    Length: 4
    Display information: sipp
  Calling party number: '1234'
    Information element: Calling party number
  Length: 6
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .010 .... = Number type: National number (0x02)
    0... .... = Extension indicator: information continues through
the next octet
    .... ..00 = Screening indicator: User-provided, not screened
(0x00)
    .01. .... = Presentation indicator: Presentation restricted
(0x01)
    1... .... = Extension indicator: last octet
    Calling party number digits: 1234
    E.164 Calling party number digits: 1234
  Called party number: '800'
    Information element: Called party number
    Length: 4
    .... 0001 = Numbering plan: E.164 ISDN/telephony numbering (0x01)
    .010 .... = Number type: National number (0x02)
    1... .... = Extension indicator: last octet
    Called party number digits: 800
    E.164 Called party number digits: 800
  User-user
```

Information element: User-user

Length: 159

Protocol discriminator: X.208 and X.209 coded user information

ACLI Instructions and Examples

You can set both of these options in the same H.323 session agent.

To set the allowCPN option for an H.323 session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
```
4. Use the ACLI **select** command so that you can work with the session agent configuration to which you want to add this option.


```
ACMEPACKET(session-agent)# select
```
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **allowCPN** with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(session-agent)# options +allowCPN
```

If you type **options allowCPN** (without the “plus” sign), you will overwrite any previously configured options. In order to append the new option to the **session-agent**’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

To set the NoPAssertedId option for an H.323 session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
```
4. Use the ACLI **select** command so that you can work with the session agent configuration to which you want to add this option.


```
ACMEPACKET(session-agent)# select
```
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **NoPAssertedId** with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(session-agent)# options +NoPAssertedId
```

If you type **options NoPAssertedId** (without the “plus” sign), you will overwrite any previously configured options. In order to append the new option to the **session-agent**’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

Trunk Group URIs

The Net-Net SBC's trunk group URI feature, applicable for SIP and IWF signaling services, enables the capabilities related to trunk groups that are described in this section. This implementation follows the IPTEL draft "Representing Trunk Groups in Tel/SIP Uniform Resource Identifiers (URIs)" (draft-ietf-ipitel-trunk-group-06.txt), and also supports more customized approaches.

- For a typical access call flow scenario, when the calling party's call arrives at the Net-Net SBC, the Net-Net SBC formulates a SIP INVITE message that it sends to a softswitch. The Net-Net SBC now supports a new URI contact parameter in the SIP request message so that service providers need to be able to:
 - Determine from where the Net-Net SBC received the call
 - Signal information about the originating gateway from a Net-Net SBC to a softswitch (e.g., an incoming trunk group or a SIP gateway to a Net-Net SBC)
- This feature supports the signaling of routing information to the Net-Net SBC from network routing elements like softswitches. This information tells the Net-Net SBC what egress route (or outgoing trunk groups) it should choose for terminating next hops/gateways. For this purpose, new SIP URI parameters in the Request-URI are defined. Additional URI parameters include the network context to identify the network in which the originating or terminating gateway resides.

- Especially important for large business applications, this feature can free Net-Net SBC resources by reducing the number of local policy, session agent, and session agent group configurations. By enabling the trunk group URI feature, the Net-Net instead uses a routing scheme based on signaled SIP URI information.

Terminology

The following IPTEL terms are used in the descriptions of and instructions for how to configure this feature:

- **Trunk**—In a network, a communication path connecting two switching systems used in the establishment of an end-to-end connection; in selected applications, it may have both its terminations in the same switching system
- **Trunk group**—A set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable except where sub-grouped
- **Trunk group name**—Provides a unique identifier of the trunk group; referred to as tgrp
- **Trunk group context**—Imposes a namespace by specifying a domain where the trunk groups are; also referred to simply as “context”

Trunk Group URI Parameters

Trunk group URI parameters identify originating and terminating trunk group information in SIP requests.

In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net SBC allows you to define URI parameters for use with originating and terminating trunk group URIs.

Originating Trunk Group URI Parameters and Formats

You can configure session agents and session agents groups on the Net-Net SBC to insert trunk group URI parameters in the SIP contact header. When SIP gateways comply with the IPTEL draft, they include the originating URI parameter in the SIP contact header. For those SIP and H.323 gateways that are not compliant, the Net-Net SBC inserts SIP trunk group URI parameters on the gateway’s behalf.

When there are no applicable session agent or session agent group configurations, the Net-Net SBC uses the source IP address of the endpoint or gateway as the trunk group name (tgrp) parameter in the originating trunk group URI.

The following diagram shows a scenario where the Net-Net inserts originating trunk group URI parameters.



There are two available formats for the originating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (identifier of the specific trunk group) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

- tgrp="trunk group name"
- trunk-context="network domain"

The URI BNF for would appear as it does in the example directly below, where the tgrp is tg55 and the trunk-context is telco.example.com:

tel:+15555551212;tgrp=tg55;trunk-context=telco.example.com

2. The second format is customized specifically for access URIs and contains two provisioned parameters: tgrp (or tgrname) and context (or provstring). This appears as tgrp.context, where these definitions apply:

- tgrp (tgrname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
- context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

custsite2NY-00020.type2.voip.carrier.net

where the tgrp is custsite2NY-00020, and the context is type2.voip.carrier.net.

The BNF for an access URI conforms to the following:

```
SIP-URI = "sip:" [userinfo ] hostport uri-parameters [headers ]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param

other-param = accessid / pname [ '=' pvalue ]
accessid = "access=" accessURI
```

```

accessURI = scheme tname [ "." provstring ]
scheme = "sip:" / token
tname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
provstring = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
domain = alphanum / alphanum *( alphanum / "-" ) alphanum

```

Terminating Trunk Group URI Parameters and Formats

Terminating trunk group URI parameters appear in the R-URI, and they can be included in by a network routing element to instruct the Net-Net SBC which egress trunk groups to use. By matching the trunk group URI parameter with configured session agents or session agent groups, the Net-Net SBC can locate the terminating gateway. The trunk group name can also be expressed as the IP address of the terminating gateway.

In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net allows you to define the URI parameters used in terminating trunk groups.

There are two available formats for the terminating trunk group URIs:

1. In compliance with the IPTEL draft, the first format has two parameters: tgrp (which can be either a trunk group name or an IP address) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:

- tgrp="trunk group name"
- trunk-context="network domain"

An example R-URI with terminating trunk group parameters appears as follows, where the tgrp is TG2-1 and the context is isp.example.net@egwy.isp.example.net:

```
INVITE sip:+15555551212;tgrp=TG2-1;trunk-
context=isp.example.net@egwy.isp.example.net SIP/2.0
```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: tgrp (or tgrname) and context (or tgdomain). This appears as tgrp.context (or tgrname.tgdomain), where definitions apply:

- tgrp (tgrname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
- context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```
SIP-URI = "sip:" [userinfo ] hostport uri-parameters [headers ]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param

other-param = egressid / pname [ '=' pvalue ]
egressid = "egress=" egressURI
egressURI = scheme tgrname [ "." tgdomain ]
scheme = "sip:" / token
tgrname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
tgdomain = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-") alphanum
domain = alphanum / alphanum *( alphanum / "-") alphanum
```

How It Works

For all trunk group URI support, you must set the appropriate parameters in the SIP manipulations configuration and in the session agent or session agent group configurations.

In the originating trunk group URI scenario, a call arrives at the Net-Net SBC from a configured session agent or session agent group. If this session agent or session agent group has the appropriate trunk group URI parameters and inbound manipulation rules configured, the Net-Net SBC then looks to the SIP manipulations configuration and add the trunk group URI information according to those rules. Those rules tell the Net-Net SBC where and how to insert the trunk group URI information, and the Net-Net SBC forwards the call.

In the terminating trunk group scenario, a call arrives at the Net-Net SBC from, for instance, a call agent. This call contains information about what trunk group to use. If the information matches a session agent or session agent group that has outbound manipulation rules configured, the Net-Net SBC will then look up the SIP manipulations configuration and strip information according to those rules. Those rules tell the Net-Net SBC where and how to remove the information, and the Net-Net SBC forwards the call.

SIP Header and Parameter Manipulation

SIP header and parameter manipulation is its own configuration where you can set up rules for the addition, removal, and modification of a SIP header or the elements of a SIP header. For example, you can set up the configuration to add a URI parameter to the URI in a SIP header or replace an FQDN with an IP address. For trunk group URI support, this configuration tells the Net-Net SBC where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

These manipulations can be applied at the realm or at the session agent level.

To learn more about SIP header manipulation, refer to the “SIP Header and Parameter Manipulation” section of this guide’s *SIP Services* chapter.

Trunk Group Routing

You can configure SIP interfaces (using the ACLI **term-tgrp-mode** parameter) to perform routing based on the trunk group information received in SIP requests. There are three options: none, IPTEL, and egress URI.

- If you leave this parameter set to none (its default), the Net-Net SBC will not look for or route based on terminating trunk group URI parameters
- When you set this parameter to either **iptel** or **egress-uri** and the incoming request has the trunk group parameter of this type (IPTEL or egress URI), the Net-Net SBC will select the egress next hop by matching the “tgrp” and “trunk context” with a configured session agent or session agent group.

If the received terminating trunk group URI parameters include an IP address, the egress next hop is the IP address specified. The Net-Net SBC determines the egress realm by matching the trunk context it receives with the trunk context you configure for the realm.

- If the incoming request does not have trunk group parameters or it does not have trunk group parameters of the type that you configure, the Net-Net SBC uses provisioned procedures and/or local policy for egress call routing.

The Net-Net SBC returns errors in these cases:

- If the terminating trunk group URI parameters do not identify a local Net-Net SBC session agent or session agent group, then the Net-Net SBC returns a SIP final response of “488 Not Acceptable Here.”
- If the Net-Net SBC receives a SIP INVITE with terminating trunk group URI parameters that do not match the specified syntax, the Net-Net SBC returns a 400 final response with the reason phrase Bad Egress=Parameters.

Trunk Group URIs and SIP Registration Caching

For calls where SIP registration caching is used, you will need to set certain parameters that enable the Net-Net SBC to preserve trunk group URI parameters on the outgoing side.

- For SIP-H.323 calls requiring IWF, you set the **preserve-user-info-sa** option in the session agent configuration.

ACLI Instructions and Examples

Before you configure your Net-Net SBC to support trunk group URIs, you need to determine:

- How you want to manipulate SIP headers (entered in the SIP header manipulations configuration)
- For terminating trunk group routing, the trunk group mode you want to use (none, IPTEL, or egress URI); this decides routing based on trunk group information
- The trunk group name and context to use entered in a session agent or session agent group configuration
- Whether you are using originating or terminating trunk group URIs (entered in the session agent configuration)
- The trunk group context for use in a realm configuration, in case the trunk group name in the session agent or session agent group does not have a context

Configuring SIP Manipulations

For detailed instructions about how to configure SIP header a manipulations, refer to the “SIP Header and Parameter Manipulation” section of this guide’s *SIP Services* chapter.

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The **new-value** parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are \$TRUNK_GROUP and \$TRUNK_GROUP_CONTEXT

Setting the Trunk Group URI Mode for Routing

To set the mode for routing for terminating trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **term-tgrp-mode**—Set the mode that you want to use. Your choices are:
 - **none**—Disables routing based on trunk groups (default)
 - **iptel**—Uses trunk group URI routing based on the IPTEL formats
 - **egress-uri**—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **out-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the Net-Net SBC via this session agent. There is no default.
5. **in-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic entering the Net-Net SBC via this session agent. There is no default.
6. **trunk-group**—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Net-Net SBC will use the one you set in the realm for this session agent.

Your ACLI entries for this list must one of these formats: **tgrp:context** or **tgrp.context**.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent)# trunk-group (tgrp1:context1 tgrp2:context2)
```

7. **options**—If you want to configure trunk group URIs for SIP-H.323 calls that use the IWF and you are using SIP registration caching, you might need to add the **preserve-user-info-sa** to your list of session agent options.

If you are adding this option to a new session agent, you can just type **options**, a <Space>, and **preserve-user-info-sa**.

If are adding this to an existing session agent, you must type a “plus” (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +preserve-user-info-sa**.

Configuring a Session Agent Group for Trunk Group URIs

In a session agent group, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent group for trunk group URIs:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-group** and press <Enter>.


```
ACMEPACKET(session-router)# session-agent-group
ACMEPACKET(session-agent-group)#
```
4. **trunk-group**—In either IPTEL or custom format, enter the trunk group names and trunk group contexts to match. If you do not set the trunk group context, then the Net-Net SBC will use the one you set in the realm for this session agent group. By default, this parameter is blank.

Your ACLI entries for this list must take one of these formats: **tgrp:context** or **tgrp.context**.

To make multiple entries, surround your entries in parentheses and separate them from each other with spaces. For example:

```
ACMEPACKET(session-agent-group)# trunk-group (tgrp1:context1
tgrp2:context2)
```

Setting a Trunk Group Context in a Realm

You can set trunk group contexts at the realm level, which will be used by all session agents and session agent groups if there is no context specified in their configurations.

The realm trunk group URI context accommodates the IPTEL and the custom format.

To configure a trunk group context for a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **trunk-context**—Enter the trunk group context to use for this realm. By default, this parameter is blank.

Using this Feature with SIP Interface Registration Caching

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the `preserve-user-info` option for that SIP interface.

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the session-related configurations.

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)# session-router
```

3. Type **session-group** and press <Enter>.

```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-interface)#
```

4. **options**—Add support for trunk group URIs with SIP interface that uses registration caching.

If you are adding this option to a new SIP interface, you can just type **options**, a <Space>, and **preserve-user-info**.

If are adding this to an existing SIP interface, you must type a “plus” (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +preserve-user-info**.

Example 1: Adding Originating Trunk Group Parameters in IPTEL Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTEL format.

```
sip-manipulation
  name add_ipTEL
  description

  header-rule
    name contact
    header-name
    action manipulate
    match-value
    msg-type any
    comparison-type case-sensitive
    new-value
    methods

  element-rule
    name tgrp
    parameter-name
    type uri-user-param
    action add
    match-val-type any
    comparison-type case-sensitive
    match-value
    new-value $TRUNK_GROUP

  element-rule
    name trunk-context
    parameter-name
    type uri-user-param
```

action	add
match-val-type	any
comparison-type	case-sensitive
match-value	
new-value	\$TRUNK_GROUP_CONTEXT

Example 1: Adding Originating Trunk Group Parameters in Custom Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in custom format.

```

sip-manipulation
  name add_att
  description

  header-rule
    name contact
    header-name
    action manipulate
    match-value
    msg-type any
    comparison-type case-sensitive

    element-rule
      name egressURI
      parameter-name
      type uri-param
      action add
      match-val-type any
      comparison-type case-sensitive
      match-value
      new-value
"sip:"+$TRUNK_GROUP+"."+"$TRUNK_GROUP_CONTEXT

```

Example 2: Removing IPTEL Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove IPTEL trunk groups names.

```

sip-manipulation
  name strip_ipTEL
  description

  header-rule
    name request-uri
    header-name
    action manipulate
    match-value
    msg-type any
    comparison-type case-sensitive

    element-rule
      name tgrp
      parameter-name
      type uri-user-param
      action delete-element
      match-val-type any
      comparison-type case-sensitive
      match-value

```

```

new-value
element-rule
  name trunk-context
  parameter-name
  type uri-user-param
  action delete-element
  match-val-type any
  comparison-type case-sensitive
  match-value
  new-value

```

Example 3: Removing Custom Trunk Group Names

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to remove custom trunk groups names.

```

sip-manipulation
  name strip_egress
  description

  header-rule
    name request-uri
    header-name
    action manipulate
    match-value
    msg-type any
    comparison-type case-sensitive

    element-rule
      name egressURI
      parameter-name
      type uri-param
      action delete-element
      match-val-type any
      comparison-type case-sensitive
      match-value
      new-value

```

Configuring SIP Manipulations

For detailed instructions about how to configure SIP header manipulations, refer to the “SIP Header and Parameter Manipulation” section of this guide’s *SIP Services* chapter.

When you configure the SIP header manipulations to support trunk group URIs, take note of:

- The name of the configuration, so that you can use it when you apply the manipulations in a session agent for the inbound or outbound manipulations
- The **new-value** parameter, which specifies the trunk group and trunk group context that you want to manipulate; the possible values that apply to trunk group URI configurations are \$TRUNK_GROUP and \$TRUNK_GROUP_CONTEXT

IWF COLP/COLR Support

When you enable the connected line identity presentation (COLP) and connected line identity restriction (COLR) feature for calls being translated between SIP and

H.323, the Net-Net SBC converts the H.323 Connected Number Information element (IE) to the SIP P-Asserted-Identity (PAI) header and vice versa.

When there is no Q.931 Connected Number IE, the Net-Net SBC converts the H.225 Connected Address alias (either E.164 or Public Party Number).

How It Works

This section describes show the IWF COLP/COLR feature works for IWF calls that originate in SIP and are translated to H.323, and for calls that originate in H.323 and are translated to SIP.

SIP to H.323 Calls

For this type of call, the Net-Net SBC checks the Connect that it receives for a Q.931 Connected Number IE. If it does not find one, then it continues by checking for H.225 Connected Address alias (either E.164 or Public Party Number). Then, it takes one of the following courses of action depending on circumstances:

- If it finds the Q.931 Connected Number IE, the Net-Net SBC extracts the screening indicator and the presentation indicator.
- If there is no Q.931 Connected Number IE, the Net-Net SBC extracts the screening indicator and the presentation indicator from the H.225 Connect-UIE of the Connect message.

With these pieces of information in place, the Net-Net SBC performs the conversion from H.323 Connected Number IE to SIP P-Asserted-Identity (PAI) header if and only if the screening indicator is either one of the following:

- **Network provided**
- **User-provided, verified and passed**

Then the Net-Net SBC adds a SIP PAI header (with URI value) to the 200 OK message that it sends in the SIP call leg. The user part of the URI is set to the value of the Q.931 Connected Number IE's numberDigits field, or to dialDigits value from the Connected Address alias. When the number type is a national number, the Net-Net SBC adds a plus sign (+) and the IWF country code (that you configure) to the beginning of the user part. If the number type is an international number, the Net-Net SBC only adds a plus sign (+). And when the Connected Number is empty, the Net-Net SBC sets the user part of the PAI header URI to **anonymous**. When the value in the presentation indicator is Presentation restricted, the Net-Net SBC adds the SIP Privacy header (with the value **id**) to the 200 OK.

In cases when it does not find a screening indicator, the Net-Net SBC will not perform the conversion from the H.323 Connected Number IE to the SIP P-Asserted-Identity (PAI) header.

H.323 to SIP Calls

For this type of call, the Net-Net SBC checks the 200 OK message for a SIP PAI header and a SIP Privacy header. Before it sends a Connect message on the H.323 call leg, the Net-Net SBC generates a Connected Number. It uses the Connected Number to insert a Q.931 Connected Number IE and an H.225 Connected Address alias (type E.164) into the Connect message. The Connected Number is generated in this way:

- If the
 - SIP PAI header is not found, or
 - User part of its URI value is **unknown** or **anonymous**, or
 - User part of its URI does not follow the H.225 NumberDigits syntax,

then the Connect Number that the Net-Net SBC generates is a Q.931 Connected Number IE that has no digits and a number type of unknown. In this case, the Net-Net SBC will not insert an H.225 Connected Address alias into the Connect message.

The presentation indicator is set to **Number not available due to interworking**, and the screening indicator to **Network provided**. The H.225 NumberDigits's syntax requires that it be between 1 and 128 characters, and only contain these characters: 0 through 9, the pound sign (#), the asterisk (*), and the comma (,).

- In all other cases, the Net-Net SBC uses the user part of the URI as the digits for the Connected Number after it performs the following:
 - Strips the plus sign in front of the number, if there is one
 - Strips the IWF country code at the beginning of the number, if there is one

Then the Net-Net SBC inserts the Connected Number into the Connect message as the Q.931 Connected Number IE and an H.225 Connected Address alias (type E.164).

If the IWF country code is found in the PAL, the Net-Net SBC sets the type of Q.931 Connected Number IE to National Number. Otherwise, the Net-Net SBC sets it to international. The screening indicator is set to **Network provided**, and the presentation indicator is set to **Presentation Restricted** if the Net-Net SBC finds a SIP Privacy header with a value of **id**, or **Presentation Allowed** if there is not SIP Privacy header.

ACLI Instructions and Examples

You configure IWF COLP/COLR support in the IWF configuration by setting two options:

- **colp-colr-iwf**—Setting this option enables support for IWF COLP/COLR
- **colp-colr-country-code**—Must be set if you configure the **colp-colr-iwf** option to recognize or build a national number; the value you enter here:
 - Must be a string of digits from 0 to 9
 - Cannot exceed 32 digits
 - Cannot contain any non-numeric characters; while it allows you to enter them, the system ignores any non-digits characters and so the feature might not work as needed

To enable IWF COLP/COLR support:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>. The system prompt will change to let you know that you can configure individual


```
ACMEPACKET(session-router)# h323
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option names with a “plus” sign in front, and then press <Enter>.

Your entry for the **colp-colr-country-code** option require that you type in the entire option name, an equal sign (=), and then the country code value.

To enter both options at once, separate the two with one command and enclose your entire entry in quotation marks (" "); see the following example for command-line syntax.

```
ACMEPACKET(h323-config)# options "+colp-colr-iwf,colp-colr-country-code=1"
```

If you type this enter without the "plus" sign, you will overwrite any previously configured options. In order to append options to the IWF configuration's options list, you must prepend the new options with a "plus" sign as shown in the previous example.

5. Save and activate your configuration.

Options for Calls that Require the IWF

You can configure several specific behaviors by configuring options for calls that require the IWF, and set them for the H.323 side of the call. These options are listed and defined in the table below. Options can be configured either globally for the H.323 configuration, individually for an H.323 interface, or for H.323 session agents.

To configure options globally for H.323:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```

From this point, you can configure H.323 parameters. To view see all H.323 parameters, enter a ? at the system prompt.
4. Type **options**, a space, and the name of the option you want to use. In this example, the **MapG729** will map H.245 G.729 to SDP G.729 with Annex B and vice versa.


```
ACMEPACKET(h323)# options MapG729
```

To configure options per individual H.323 interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
```
4. Type **h323-stack** and press <Enter>. The system prompt changes again to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

From this point, you can configure H.323 interface parameters. To view see all H.323 interface parameters, enter a ? at the system prompt.
5. Type **options**, a space, and the name of the option you want to use. In this example, the **MapG729** will map H.245 G.729 to SDP G.729 with Annex B and vice versa.


```
ACMEPACKET(h323-stack)# options MapG729
```

To configure options for H.323 session agents:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```

3. Type **session-agent** and press <Enter>.

```
ACMEPACKET(session-router)# session-agent
```

From this point, you can configure session agent parameters. To view see all session agent parameters, enter a ? at the system prompt.

4. Type **options**, a space, and the name of the option you want to use. In this example, the **MapG729** will map H.245 G.729 to SDP G.729 with Annex B and vice versa.

```
ACMEPACKET(h323-stack)# options MapG729
```

Options	Description
MapG729	Net-Net SBC maps H.245 G.729 to SDP G.729 with Annex B and vice versa. Applicable only to calls that require the IWF.
ColonG729	Net-Net SBC uses the : (colon) instead of the = (equal sign) in the media attribute line a=fmtp:18 annexb=yes/no when mapping H.245 G.729 or SDP G.729 with Annex B. Applicable only to calls that require the IWF.
IwfLRQ	Net-Net SBC sends an INVITE (with no SDP) to a redirect server in response to an incoming LRQ received on an H.323 interface. If a 3xx message with a redirected contact header is returned, the Net-Net SBC will send an LCF in response to the LRQ. Otherwise, it will send an LRJ.
NoG729AnnexB	SDP received by the IWF with H.729 and no FMTP will be mapped to G.729 on the H.323 side of the call. Can also be set in the session agent options parameter.
sameT38Port	Net-Net SBC's H.323 process does not allocate separate ports for audio and T.38. Net-Net SBC will send the same audio port in the OLCAck that it sees in a request mode for T.38 and a new OLC for T.38.
pvtStats	Net-Net SBC includes program value tree (PVT) statistics in the show h323d display that are a sum of the PVT statistics for all H.323 interfaces. Used for debugging purposes.
acceptAI	Net-Net SBC accepts all the codecs received in the SIP 200OK and builds the TCS accordingly.

Suppress SIP Reliable Response Support for IWF

For IWF-originated calls, the Net-Net SBC now allows you to configure the suppression of the SIP 100rel option tag on a per-H.323 interface (stack) basis.

When a call originates on the H.323 side for a call that requires interworking between H.323 and SIP, the Net-Net SBC inserts the 100rel option tag in the Supported header of the outgoing SIP INVITE. Although this behavior is required for RFC 3262 conformance, and is ignored by endpoints that do not support this RFC, suppressing the reliable response can alleviate processing burdens and avoid the possibility that an endpoint could mishandle the response.

In addition, enabling this feature suppresses the same 100rel options tag in the Required header for outgoing IWF responses for which an incoming SIP INVITE had that same tag in its Supported header. If an incoming INVITE requires reliable provisional responses and the SIP feature configuration is set to accept the 100rel, the Net-Net SBC then includes the 100rel option tag in the outgoing response's Required header. When the SIP feature is not so configured, the Net-Net SBC rejects the INVITE with a 420 Bad Extension response.

Without this option, you can suppress the reliable response on a global basis or per SIP next-hop by using the SIP feature configuration. However, using this feature allows a finer degree of granularity by making the functionality only applicable to IWF calls that originate in H.323.

ACLI Instructions and Examples

To suppress the SIP 100rel option tag:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>.


```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI **select** command) the configuration that you want to edit.
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **suppress100rel** with a “plus” sign in front of it, and then press <Enter>.


```
ACMEPACKET(h323-stack)# options +suppress100rel
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

IWF Codec Negotiation: H.323 Slow Start to SIP

For instances when the Net-Net SBC is translating a call initiated in H.323 slow start to SIP, you can enable a setting in the IWF configuration that prevents the sending an SDP offer in the SIP INVITE. Instead, the Net-Net SBC expects to see an SDP offer from the SIP endpoint in a provisional or reliable/provisional 200 OK, and then sends an answer in an ACK or PRACK.

With this parameter disabled (default), the Net-Net SBC populates the SIP INVITE with SDP based on the media profiles applied to the ingress H.323 session agent or the IWF configuration.

ACLI Instructions and Examples

To prevent the Net-Net SBC from sending an SDP offer in the SIP INVITE for a call being translated between H.323 slow start and SIP:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **iwf-config** and press <Enter>.


```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```
4. **slow-start-no-sdp-in-invite**—Change this parameter to enabled if you want to prevent the Net-Net SBC from sending an SDP offer in the SIP INVITE for an IWF call initiated in H.323 slow start (being translated to SIP). The default is disabled.
5. Save and activate your configuration.

IWF: H.245 Signaling Support for G.726

In addition to providing G.726 support for pure SIP and pure H.323 calls, the Net-Net SBC supports the G.726 payload type for H.245 and calls that require interworking (IWF) between SIP and H.323.

How It Works: IWF

For IWF calls using ITU-T G.726 as the audio codec, the SIP call leg requires G.726 in the SDP. The H.323 side of the call signals G.726 (in the H.245 openLogicalChannel and TerminalCapabilitySet messages) by including a GenericCapability defining G.726 as the codec. In the GenericCapability, the capabilityIdentifier and maxBitRate parameters identify G.726. While a capabilityIdentifier with 0.0.7.726.1.0 designates G.726, the maxBitRate designate the data transmission rate.

Codec	Max Bit Rate	Data Rate
G726-16	160	16 kbit/s
G726-24	240	24 kbit/s
G726-32	320	32 kbit/s
G726-40	400	40 kbit/s

To support G.726 for IWF calls, the Net-Net SBC converts the G726-X value in the SDP of SIP messages to a GenericCapability structure in H.323/H.245 messages, and the conversion works the same way in reverse.

ACLI Instructions and Examples

To enable this feature, you do need to set up media profile configurations appropriately. Media profiles now allow you to set the configuration to any of the four G.726 encodings (as defined by ITU G726 Annex B and RFC 3551). You must create one media profile for each of the four different supported data rates. In addition, you are also required to set a genericAudioCapability media profile.

To set a media profile for H.245 and IWF G.726 support:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **media-profile** and press <Enter>.

```
ACMEPACKET(session-router)# media-profile
```

```
ACMEPACKET(media-profile)#
```

4. **name**—Set the name of the media profile to G726-16. Values to support this feature are: G726-16, G726-24, G726-32, and G726-40.
5. **media-type**—Set the media type to use for this media profile; for generic video, set this parameter to **audio** (default)
6. **payload-type**—Set the payload type to use for the generic video media profile.
7. **transport**—Set the transport type to use for the generic video media profile; the default is RTP/AVP.
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a media profile configuration for H.245/IWF G.726 support:

```
media-profile
  name                g726-40
  media-type          audio
  payload-type        105
  transport            RTP/AVP
  req-bandwidth        0
  frames-per-packet    0
  parameters
  average-rate-limit    0
  sdp-rate-limit-headroom 0
  sdp-bandwidth         disabled
```

To set a media profile for generic audio support:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **media-profile** and press <Enter>.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **name**—Set the name of the generic audio media profile to **genericAudioCapability**. There is no default for this parameter.
5. **media-type**—Set the media type to use for this media profile; for generic video, set this parameter to **audio** (default)
6. **payload-type**—Set the payload type to use for the generic audio media profile.
7. **transport**—Set the transport type to use for the generic audio media profile; the default is RTP/AVP.
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic audio media profile configuration:

```
media-profile
  name                genericAudioCapability
```


media-type	audio
payload-type	104
transport	RTP/AVP
req-bandwidth	0
frames-per-packet	0
parameters	
average-rate-limit	0
sdp-rate-limit-headroom	0
sdp-bandwidth	disabled

International Peering with IWF and H.323 Calls

In previous Net-Net SBC releases, SIP to H.323 IWF calls default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option **cpnType=X**, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outgoing calls on the h323-stack.

In addition to SIP to H.323 IWF calls, the **cpnType=X** option is used to set the Q.931 Number Type for H.323-H.323 calls.

The following is a list of possible values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

ACLI Instructions and Examples

You configure this feature as an option in the **h323-stack** configuration.

To configure the cpnType=X option for H323-H323 calls:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **h323-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323)#
```
4. Type **h323-stacks** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```

If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration you want to edit.

5. Set the options parameter by typing **options**, a <Space>, the option name **cpnType=x** (where x is the Q.931 Number Type) with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(h323-stack)# options +cpnType=1
```

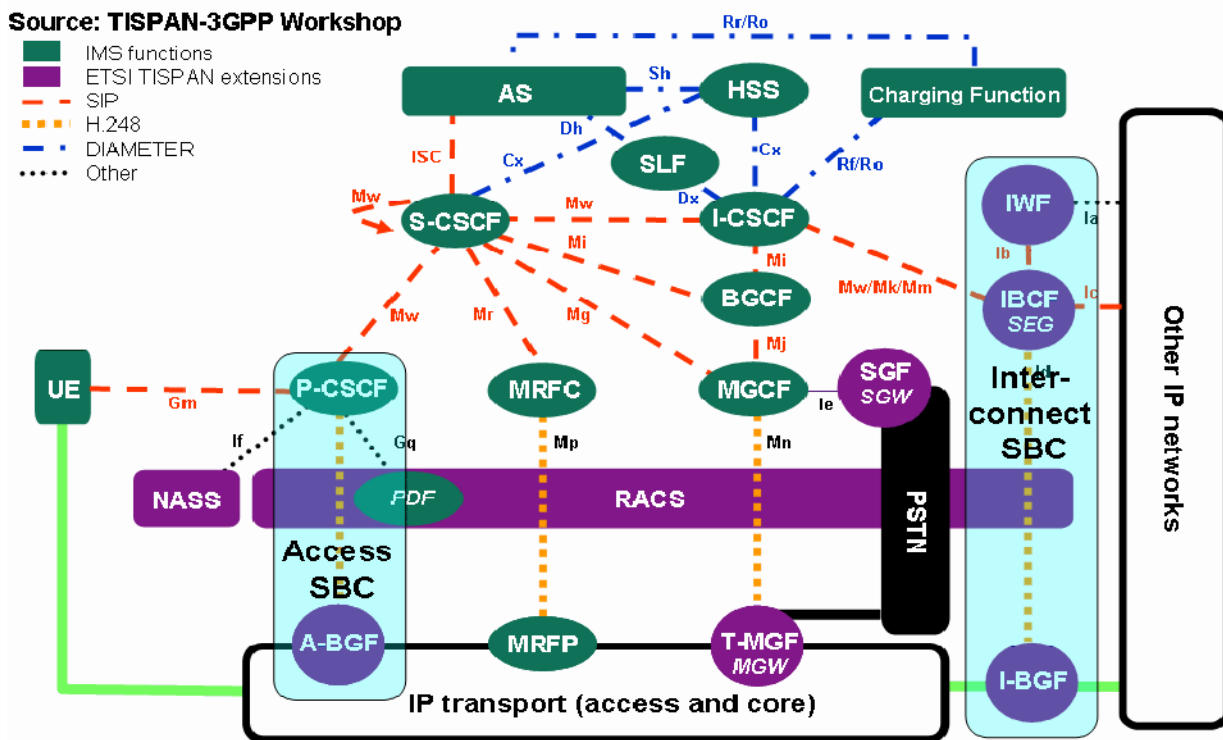
If you type **options** without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the h323-stack’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

Net-Net 9200 IMS Support

The ETSI TISPAN NGN defines several subsystems that make up the NGN architecture. The model for the target NGN architecture is depicted below. Acme Packet's Net-Net Session Director is an integrated session control, policy enforcement and media management solution that incorporates functional components of the IP multimedia subsystem (IMS) the Resource and Admission Control Subsystem (RACS) and functions necessary for the interconnection with other IP networks/domains. The functions of the Net-Net 9200 within the NGN architecture are divided into the interconnect border functions and the access border functions. The diagram below depicts the mapping of these functions across IMS architecture.

Source: TISPAN-3GPP Workshop



Net-Net 9200 Access Border Functions

- Proxy CSCF (P-CSCF)
- Access/Core Border Gateway Function (A/C-BGF)
- RACF AF and SPDF functions

Net-Net 9200 Interconnect Border Functions

- Interconnect Border Control Function (I-BCF)
- Interconnect Border Gateway Function (I-BGF)

IMS Access Border Functions

The Net-Net 9200 is deployed as the access point between the core IMS network and UEs to deliver the functions defined in the TISpan architecture as the P-CSCF and A-BGF. These two functions can not be separated. The Net-Net 9200 performs the functions describe in this section as the Access SBC.

P-CSCF Functions

The Net-Net 9200 performs the following functions in the role of P-CSCF:

- Forwards SIP REGISTER messages and maintains a cached mapping of the user info and the UE's Address of Record (AoR), including the far-end NAT address in the case of hosted NAT traversal (HNT).
- Forwards SIP messages to a S-CSCF based on service route discovery procedures.
- Performs local emergency session handling—Local routing policy is used by the Net-Net 9200 to identify emergency sessions and provide unique routing (e.g. can route to a dedicated S-CSCF function for emergency session handling).
- Operates as a UA (B2BUA) for generating independent SIP transactions for security purposes and handling of abnormal conditions.
- Offers current session timers which are used to monitor for media faults and abandoned calls.
- Generation of CDRs—The Net-Net 9200 generates real-time accounting records via RADIUS.
- Authorization of bearer resources and QoS management—With integrated BGF capabilities, the Net-Net 9200 allocates bearer resources (NAPT flows) and applies QoS policies (including packet marking) based on local policies.
- Traffic Policing—Traffic is policed at the session and media/transport layer. At the signaling layer, the Net-Net 9200 polices at a number of levels including:
 - Capacity—Total number of concurrent calls to/from each realm
 - Session set-up rate—Maximum rate of call attempts to/from each signaling element
 - Signaling message rate—Each endpoint's signaling message rate is monitored and policed
 - Signaling bandwidth—each endpoint's signaling bandwidth is policed individually

SBC Protecting the P-CSCF

When the Net-Net 9200 itself is not the P-CSCF, it can be used to specifically protect the P-CSCF signaling layer in the IMS/TISpan model. To accommodate for this functionality for some implementations, the Net-Net 9200 provides the following behaviors:

- Implicit Service-Route
- Modified P-CSCF Redundancy
- Tracking UE per P-CSCF
 - Received Header Contact Encoding

A-BGF Functions

The Net-Net 9200 performs the following IMS BGF functions:

- Opening and closing gates/packet filtering—The Net-Net 9200 opens and closes gates (media pinholes) on a session-by-session basis. Packet filtering rules include full source and destination IP address and port number.
- Per-session DiffServ or ToS marking—Media flows destined for the IMS core network can be explicitly marked using ToS or DiffServ. Media packets can be marked by VPN, by codec (voice, video) or by E.164 phone number prefix.
- NAPT-PT and topology hiding—The Net-Net 9200 provides NAPT for all media flows associated with a session on a per session-basis. Double NATing, NATing both source and destination sides, is utilized to fully hide topology in each direction for RTP and RTCP. Local IP addresses and port resources are dynamically allocated from steering pools provisioned on the Net-Net 9200.
- Hosted NAT traversal—The Net-Net 9200 supports HNT function that allows media flow traversal through the CPE firewall/NAT without upgrading the CPE equipment. The Net-Net 9200 interacts with the endpoints to dynamically establish and maintain bindings in the CPE firewall/NAT that allow the signaled communications to pass through. The Net-Net 9200's registration management and media relay functions make CPE-based NATs transparent to the service delivery elements.
- Traffic Policing—Traffic is policed at the session and media/transport layer. At the signaling layer, the Net-Net 9200 polices at a number of levels including:
 - Policing of Media (e.g. RTP & RTCP) traffic on a per-flow basis—CBR policing is applied to each flow based on negotiated offered and negotiated media codecs.

IMS Interconnect Border Functions

The Net-Net 9200 is deployed at IP interconnect points between service providers to deliver the functions defined in the TISpan architecture as the I-BCF and I-BGF. The Net-Net 9200 performs the following functions as the Interconnect border SBC.

Interconnect Border Control Function (I-BCF)

- Interaction with I-BGF (including NAPT and firewall functions)
- Topology hiding—screening of signalling information

Interconnect- Border Gateway Function (I-BGF)

- Gate opening/closing
- NAPT and packet filtering
- Packet marking
- Resource allocation and bandwidth reservation
- Security and topology hiding
- Session admission control, resource and traffic management
- Upstream/downstream flow policing
- Quality monitoring and reporting
- Usage metering - CDR generation

IMS Path and Service Route Header Support

The Net-Net 9200 supports the Path header and the Service-Route header used in the registration phase of a SIP transaction. The Net-Net 9200 will learn the route vectors from the SIP URIs contained in these headers in order to preload SIP headers with the correct route vectors in subsequent SIP message exchanges between the UA and the S-CSCF across the Net-Net 9200. This is how the Net-Net 9200 supports RFC 3608 and RFC 3327.

Path Header

When a UE registers to an S-CSCF, the Net-Net 9200 adds the Path header in the REGISTER message as it is proxied to the S-CSCF. The Path header includes the SIP URIs that form the route vector which describes how the UE reaches the Net-Net 9200, through a specific series of proxies. This route vector is saved in the Net-Net 9200's registration entry for the UE, routing all subsequent SIP messages from the S-CSCF to the UE. As the Path header is sent to the S-CSCF, the Net-Net 9200, as P-CSCF, inserts the SIP URI of itself as the top entry in the Path header.

The Path header only appears in SIP messages exchanged during the registration process.

If the REGISTER request already contains a Path header, the Net-Net 9200 stores the contents of the Path header(s) for this endpoint for routing back to the endpoint in subsequent messages.

Service Route Header

When a UE registers through the Net-Net 9200 to the registrar, the registrar returns a Service-Route header in a 200 OK message in response to the REGISTER message to the UE. This header contains the route vector that directs traffic through a specific sequence of proxies used to reach the S-CSCF. The Service-Route header only appears during the SIP registration process.

The P-CSCF (Net-Net 9200) will now store the URIs listed in the Service-Route header(s) in the registration entry of the UE for use in routing subsequent traffic to the S-CSCF. The Net-Net 9200 inserts this sequence of proxies into and outgoing message's Route headers; this is called a pre-loaded route. This route is only applicable for the traffic flowing between the originating UE and the contacted S-CSCF.

When receiving subsequent requests from the UE, the Net-Net 9200 looks at the UE's registration entry for a service route, and will insert the route vector as appropriate Route headers. If the service route is not found in the registration entry, the routing is performed in the usual fashion.

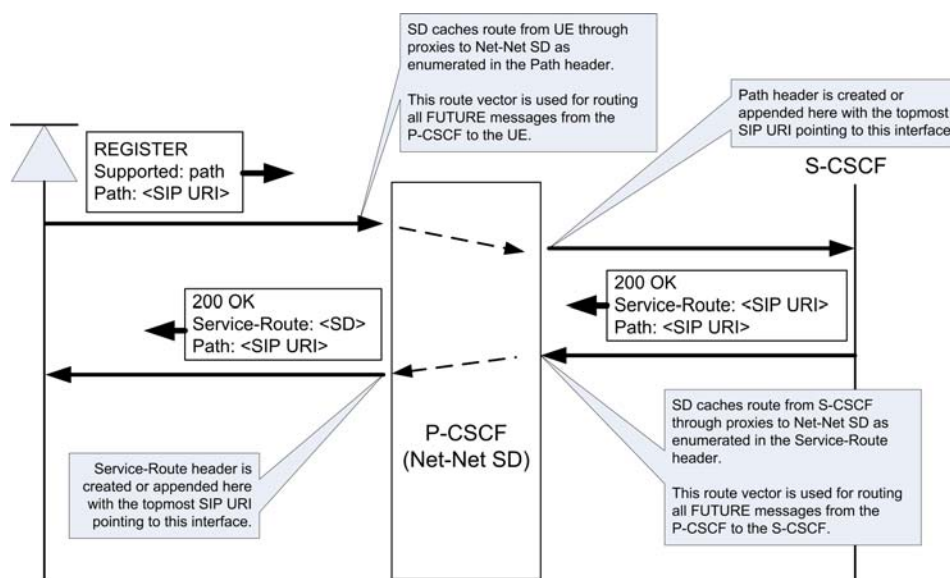
As an exception, you may wish for the Net-Net 9200 to not use the Service-Route header to route subsequent Register requests. Note in the configuration section the way to disable Service-Route header routing.

The manner in which passing or stripping of Service-Route headers sent from the S-CSCF is done is determined by local configuration on the Net-Net 9200. There is no verification of configured local policy against the route included in the stored service route. The Service-Route header, as created by the Net-Net 9200, and exiting back to the UE, contains a SIP URI pointing to itself as the topmost entry. This is used so that other proxies can learn the route back to the Net-Net 9200.

Summary

If a request originates at the UE, the routes enumerated in the Service-Route header are used to route the request to the S-CSCF. If a request is meant to terminate at a UE, the routes enumerated in the Path header are used to route the response to the UE. Service-Route routes take priority over configured local policy.

Path headers received in a 200 OK response from the registrar are transmitted to the UE unchanged. If you want them stripped as the SIP message leaves the Net-Net 9200, you can use the SIP Header Manipulation function.



ACLI Configurations and Instructions

IMS and all related functions must be enabled on both the access-side and core-side SIP interfaces. Only IMS features discussed up to this point are enabled by the following procedure.

To enable RFC 3608 and RFC 3327 support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **select** and the number of the pre-configured sip interface you want to configure.


```
ACMEPACKET(sip-interface)# select 1
```
5. Type **sip-ims-feature enabled** and press <Enter>. This completes enabling IMS for a given SIP interface.

```
ACMEPACKET(sip-interface)# sip-ims-feature enabled
```

If you wish to disable subsequent routing of REGISTER messages via the Service-Route header, type options followed by a <Space>, the plus sign and **route-register-no-service-route**. Then press <Enter>.

6. Save and activate your configuration.

Service Route Enhancements

This section describes the enhancements to support for the Service Route header for SIP. These reference point for how the new enhancements work is based on Net-Net 4000 behavior.

These changes provide flexibility to cover possible uses cases where there have previously, in Net-Net 4000 configuration and operation, been the following types of issues:

- Even when IMS is disabled for a SIP interface, the Net-Net 4000 caches and uses the Service-Route headers from SIP REGISTER responses the registrar returns. In that configuration, you must use SIP HMR to remove the Service-Route headers from the response, and there is no mechanism for replacing the Service-Routes from the REGISTER response with an implicit Service Route.
- In the Net-Net 4000 global SIP configuration, the presence of the option **route-registrar-no-service-route** sets the behavior for using the Service-Route header in the REGISTER request. This option has multiple parameters to define very specific behavior, so the new enhancements greatly simplify the process of determining proper use of the header in both IMS and non-IMS environments.
- You can configure the Net-Net 4000 with an option to keep it from using the Service-Route header for REGISTER requests when sent to an out-of-service session agent. The enhancements make this behavior the default—because otherwise these REGISTER requests fail.

SIP Interface: Implicit Service Route

There are two new values, **absent** and **replace**, you can set for the **implicit-service-route** parameter in the SIP interface configuration.

- **disabled** (default)—Turns off the implicit service route feature; Net-Net SBC constructs service route the Service-Route headers in a successful REGSITER response.
- **enabled**—Turns on this feature, meaning that an implicit service route to the session agent to which the REGISTER request was sent is inserted in front of Service-Route header in a successful REGISTER response; the inserted URI includes the **;lr** parameter if the session agent has loose routing enabled.
- **strict**—An implicit service route to the session agent to which the REGISTER request was sent is inserted in front of Service-Route header in a successful REGISTER response; the inserted URI does not the **;lr** parameter if the session agent has loose routing enabled, overriding the loose routing behavior configured for the session agent.
- **absent**—An implicit service route to the session agent to which the REGISTER request was sent is constructed when the successful REGISTER response contains no Service-Route headers.
- **replace**—An implicit service route to the session agent to which the REGISTER request was sent is used to construct the service route. The Net-Net SBC ignores Service-Route headers in successful REGISTER responses.

ACLI Instructions and Examples

To choose how you would like the Net-Net SBC to construct the implicit service route:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **implicit-service-route**—Enter how you want the Net-Net SBC to construct implicit service routes. Valid values are: **disabled** (default), **enabled**, **strict**, **absent**, and **replace**.
5. Save and activate your configuration.

Service Route Use

Once an endpoint registers successfully, the Net-Net SBC caches the Service-Route header (if any) to use for routing all subsequent requests from the endpoint—with the exception of any subsequent REGSITER requests.

You can set whether or not you want the Net-Net SBC to route subsequent REGISTER requests using the cached Service Route, and whether the endpoint is engaged in an active session through the Net-Net SBC. If you decide not to use the Service Route for endpoints engaged in active sessions, then the Net-Net SBC uses the local policy to make routing decisions.

For endpoints not in found in the Net-Net SBC's registration cache, the Net-Net SBC again uses the local policy to make routing decisions.

ACLI Instructions and Examples

To configure how you want the Service Route used:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **register-service-route**—Enter the way you want the Net-Net SBC to use the service route:
 - **always**—The Net-Net SBC always uses the cached service route for an endpoint for routing REGISTER requests.
 - **never**—The Net-Net SBC never uses the service route, and makes routing decisions based on local policies instead.

- **removal**—The Net-Net SBC uses the cached service route for an endpoint when routing REGISTER requests that remove the endpoint's contact. It uses the local policy for refresh and query REGISTER requests.
- **session**—The Net-Net SBC uses the cached service route when routing REGISTER requests that appear while an endpoint has an active session traversing it. When an endpoint does not have an active session, the Net-Net SBC uses the local policy to make routing decisions.
- **session+removal**—Combining the **session** and **removal** values, the Net-Net SBC uses the cached service route: when routing REGISTER requests that remove the endpoint's contact and when REGISTER requests appear while an endpoint has an active session traversing the system. Otherwise, the Net-Net SBC uses the local policy to make routing decisions.

5. Save and activate your configuration.

Notes About Upgrading

There are Net-Net SBCs currently deployed that use the **route-registrar-no-service-route** option, and these enhancements provide for backward compatibility.

When you upgrade to a release that has the new **register-service-route** parameter in the SIP configuration, the system checks for the presence of the **route-registrar-no-service-route** option. If the system finds the option, then it translates the value configured for the option like this:

Old route-registrar-no-service-route value	New register-service-route value
<empty> or all	never
refresh	removal
all, idle	session
refresh; idle	session+removal

You must save your configuration for these changes to take place, allowing you to fall back to the previous software image.

P-CSCF Endpoint Identification Using Address and Port

You can configure the Net-Net SBC, acting as a P-CSCF, to match a Request it receives to a registration cache entry based only on the IP address and port from which the Request came. When you enable this behavior, the Net-Net SBC will perform this kind of endpoint identification even when there nothing in the message matches the cache entry.

ACLI Instructions and Examples

For this behavior to work as designed, you must also have the **reg-via-key** option enabled for the SIP interface to which you are adding the **reg-via-match** option.

To configure P-CSCF endpoint identification using address and port:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>. If you are editing an existing configuration, select the one on which you want to enable this feature.

```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-interface)#
```

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **reg-via-match** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(sip-interface)# options +reg-via-match
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

IMS Support for Private Header Extensions for 3GPP

As part of its RFC 3455 support, the Net-Net 9200 supports the following headers in its IMS implementation:

- P-Associated-URI
- P-Asserted-Identity
- P-Called-Party-ID
- P-Charging-Function-Address
- P-Visited-Network-ID

The procedure to enable IMS support is explained under ACLI Configurations and Instructions in the previous section. IMS and all related functions must be enabled on both the access-side and core-side SIP interfaces.

P-Associated-URI Header

In the SIP registration process, the registrar often returns a set of associated URIs for a registering AoR. When the Net-Net 9200 receives the list of associated URIs, it stores them in the registration entry for the registering endpoint. The service provider allocates one or more associated URIs per user for his or her own usage. After an endpoint successfully registers, the P-Associated-URI header returned in a 200 OK message informs the UE of all URIs associated with the AoR.

When the Net-Net 9200 receives a request from a UE, the URI in the From header is matched against the registration cache for that endpoint. If the registering endpoint matches an associated-URI already in the registration table, the Service-Route associated with this endpoint is used to create the route for originating transactions associated with the endpoint to the S-CSCF.

The inclusion or exclusion of the P-Associated-URI header is not dependent on the trust level of an ingress or egress realm.

P-Asserted-Identity Header

The Net-Net 9200 inserts a P-Asserted-Identity header into any initial request for a dialog or standalone transaction sourced by the UE.

The inclusion or exclusion of the P-Asserted-Identity header is dependent on the trust level of an egress realm.

P-Asserted-Identity Header Handling

1. The Net-Net 9200 inserts a P-Asserted-Identity header into all messages other than the REGISTER message.
2. When the P-Preferred-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is also present in the UE's associated URI list, then this SIP URI is inserted in the P-Asserted-Identity header as the SIP message enters the core network.
3. When the P-Asserted-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is also present in the UE's associated URI list, then the original P-Asserted-Identity header and SIP URI is passed unchanged into the core network.
4. When the From header is present in an INVITE sourced by the UE, and the SIP URI contained in this header appears in the UE's Associated URI list, then this SIP URI is inserted into the P-Asserted-Identity header as the SIP message enters the core network.
5. When the P-Asserted-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is not present in the Associated URI list, the Net-Net 9200 acts like no P-Asserted-Identity was received from the UE.
6. When no P-Asserted-Identity can be derived from an INVITE sourced by the UE, the P-Asserted-Identity is based on the first URI in the Associated URI list.
7. The P-Asserted-Identity header will be removed from SIP messages sent and received from a UE if either the ingress or egress side is untrusted and the UE's Privacy header's contents is "id".
8. If no P-Associated-URI exists for a registered endpoint, the Net-Net 9200 will use the configured default P-Asserted-Identity found on the sourcing session agent.
9. If the session agent that originates a message does not include a P-Asserted-Identity header or the request is not originated from the session agent, and the P-CSCF has not received P-Associated-URI list from the registrar for a particular user, no P-Asserted-Identity will be created.
10. The P-Preferred-Identity header will never be passed to the S-CSCF.

If the above steps fail to insert a P-Asserted-Identity header, you can manually configure a value to be inserted into a P-Asserted-Identity header. The **sip-ims-feature** parameter must still be enabled to use the P-Asserted-Identity header override.

ACLI Instructions and Examples

P-Asserted-Identity header handling is enabled with the sip-ims-feature as described in the previous section.

A P-Asserted-Identity header can be manually configured for a session agent if the automatic logic, explained above in this section, fails.

To configure the P-Asserted-Identity header for a session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


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

```
ACMEPACKET(session-router)# session-agent
```

```
ACMEPACKET(session-agent)#
```

4. Type **select** and the number of the pre-configured session agent you want to configure.

```
ACMEPACKET(session-agent)# select 1
```

5. **p-asserted-id**—Enter the URI to use when no P-Asserted-Identity has been created.

```
ACMEPACKET(session-agent)# p-asserted-id sip:name@acmepacket.com
```

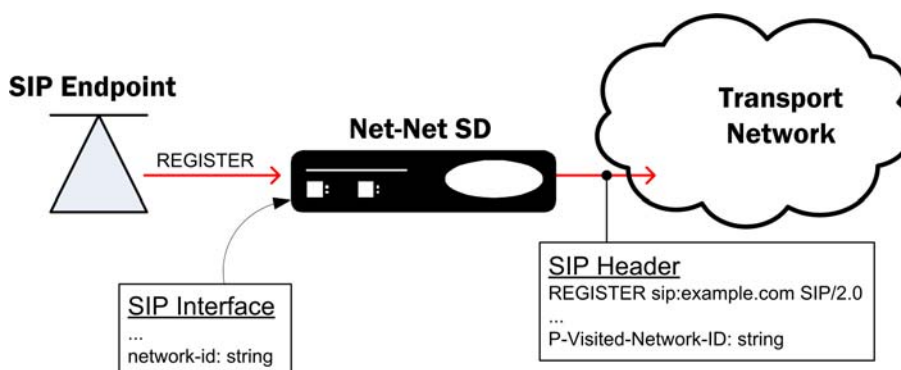
6. Save and activate your configuration.

P-Called-Party-ID Header

The Net-Net 9200 transparently passes the P-Called-Party-ID header between the S-CSCF and a UA. The inclusion or exclusion of the P-Called-Party-ID header is dependent on the trust level of the egress realm.

P-Visited-Network-ID Header

The Net-Net 9200's IMS support also includes the insertion of a P-Visited-Network-ID header into SIP messages when applicable. When a UE sends a dialog-initiating request (e.g., REGISTER or INVITE message) or a standalone request outside of a dialog (e.g., OPTIONS) to the P-CSCF, the Net-Net 9200 inserts the P-Visited-Network-ID header into the SIP message as it enters into the destination network.



The P-Visited-Network ID header will be stripped from SIP messages forwarded into untrusted networks as expected. The content of a P-Visited-Network-ID header is a text string that identifies the originating UE's home network. This string is user-configurable.

ACLI Configurations and Instructions

P-Visited-Network-ID header handling is enabled with the sip-ims-feature as described earlier. The actual P-Visited-Network-ID string must be configured on the access-side SIP interface.

To configure the P-Visited-Network-ID string in a SIP interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the session-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Type **select** and the number of the pre-configured sip interface you want to configure.

```
ACMEPACKET(sip-interface)# select 1
```

5. **network-id**—Enter the P-Visited-Network-ID string for the SIP interface.

```
ACMEPACKET(sip-interface)# network-id examplenetworkid
```

6. Save and activate your configuration.

P-Asserted-ID Output to CDRs

The Net-Net SBC does not write VSA 69, Acme-P-Asserted-ID, if the following two conditions are true:

1. The SIP message's destination is untrusted
2. Privacy is on. i.e., an id is contained within the Privacy: header.

In order to override this behavior and write the P-Asserted-ID's header value to a CDR, enable the **insert-pai** option in the account config.

ACLI Configuration

To configure the Net-Net 9200 to include P-Asserted-IDs in CDRs:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **account-config** and press <Enter>.

```
ACMEPACKET(session-router)# account-config
ACMEPACKET(account-config)#
```

4. Set the options parameter by typing **options**, a <Space>, the option name **insert-pai** with a "plus" sign in front of it, and then press <Enter>.

```
ACMEPACKET(account-config)# options +insert-pai
```

If you type **options** without the "plus" sign, you will overwrite any previously configured options. In order to append the new options to the account-config's options list, you must prepend the new option with a "plus" sign as shown in the previous example.

5. Save and activate your configuration.

IMS Charging Headers

The Net-Net 9200 supports IMS Charging Headers. These headers include P-Charging-Vector and the P-Charging-Function-Address. IMS charging header support is configured separately from other IMS functions in order to support a variety of customer needs. Charging header information is recorded in the CDR records.

A charging vector is defined as a collection of the charging information defined in RFC 3455. It is used to correlate charging records among network elements. The

charging vector is constructed during the establishment of the dialog or a standalone transaction outside of a dialog.

The Net-Net SBC's Charging header handling has been designed around the assumption that charging headers are intended for the IMS core only and should be readily eliminated from the access realms. The Net-Net SBC's charging header implementation can add, pass through, or remove relevant headers to accommodate for a variety of roles the Net-Net SBC may play in IMS deployments.

When charging headers are forwarded from an egress interface, one of two things can happen:

- The Net-Net SBC can be configured to forward the charging headers' contents as received on the ingress interface
- The Net-Net SBC can be configured to insert pre-configured or previously received values in the charging headers and forward them into the egress realm.

P-Charging Vector Handling

The P-Charging-Vector header is composed of four parameters: icid-value, icid-generated-at, orig-ioi, term-ioi. See RFC 3455, Section 4.6 for more information. You can preform several actions on the charging header by configuring the **charging vector mode** parameter on the egress interface as follows:

- **pass** – Forwards charging headers through the Net-Net SBC, unmodified. The Net-Net SBC saves the P-Charging-Vector's icid-value for accounting (CDRs) purposes. This mode is appropriate when the Net-Net SBC lies completely within the IMS core, and not on the edge.
- **insert** – Removes any existing charging header values and insert new ones from configuration or those values that were previously saved (except values saved in the registration cache). This mode is appropriate when the Net-Net SBC lies on the edge of the IMS network. The insert action is configured on the access SIP interface and is normally paired with delete mode in the corresponding, core SIP interface.
- **delete** – Deletes any charging header received. This mode is appropriate to use when the Net-Net SBC lies on the edge of the IMS network. The delete action is configured on the IMS core SIP interface and is normally be paired with the insert mode on the corresponding access SIP interface.
- **delete-and-respond** – Delete and charging header received, but save out-of-dialog values to insert in future in-dialog messages. This mode is appropriate to use when the Net-Net SBC lies on the edge of the IMS network. The **delete and respond** action is configured on the IMS core SIP interface and is normally be paired with **insert reg cache** (PCFA) mode or insert (for PCV) mode in the corresponding access sip-interface.

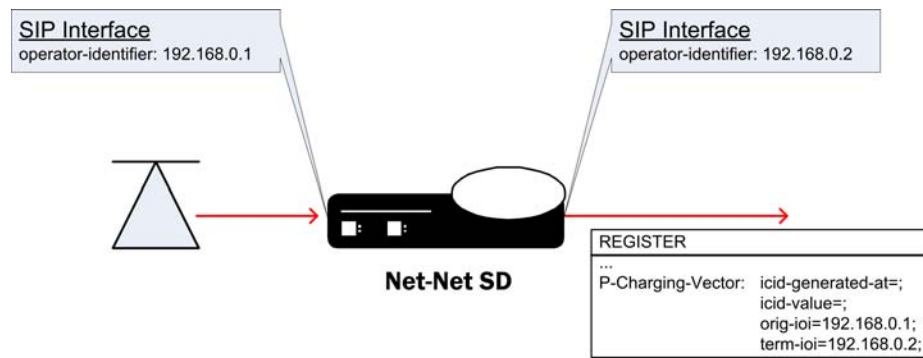
When inserting a P-Charging-Vector header, the Net-Net SBC checks if there are existing parameters for the current session. Saved values are used instead of generating new ones when in insert mode. If there is no saved icid-value, it must be generated.

1. The Net-Net 9200 constructs a globally unique icid-value.
2. The icid-generated-at parameter's value is the IP address of the egress SIP interface. This value is generated by the Net-Net 9200.
3. The orig-ioi parameter's value is set manually using the operator-identifier field located in the SIP interface configuration element of the originating network. If the parameter has already been saved in the session, then that value is used.

- The term-ioi parameter's value is set manually using the operator-identifier field located in the terminator's SIP interface configuration element. This operates exactly the same as the orig-ioi parameter, except that it is NOT sent or constructed on any initial request. Since this is a "terminator" value, it comes from the response from the terminator. Values saved in the session are used if present, but otherwise the parameter would only be generated on a response. Example

```
P-Charging-Vector: icid-
value=1ate6g46n1823s8719ck3ps6gbt46m5d3bci3po5hhdg3n86g1csio47g9c43;
icid-generated-at=192.168.0.2;
orig-ioi=192.168.0.1;
term-ioi=192.168.0.2;
```

Based on the direction of the call, the Net-Net 9200 will insert the operator-identifier parameter into the orig-ioi and the term-ioi parameters. The orig-ioi parameter takes the value of the operator-identifier configuration parameter of the SIP interface that receives the SIP request. The term-ioi parameter takes the value of the operator-identifier configuration parameter of the SIP interface that sends the SIP request to its next hop.



P-Charging-Function-Address

The P-Charging-Function-Address header is comprised of two parameters: ccf, ecf. It can have multiple instances of these parameters. As such, multiple addresses can be specified in the **ccf address** and **ecf address** parameters in the SIP interface when separated by commas.

When inserting a P-Charging-Function-Address header on egress, the values used are chosen in the following order:

- P-Charging-Function-Address saved in session
- For responses, P-Charging-Function-Address from the corresponding request if the charging header mode on the incoming interface is **delete and respond** and the message lacks a To: tag, which would indicate that it's an out-of-dialog response.
- P-Charging-Function-Address saved in registration cache for the registered or target user. This is only applicable for the **insert reg cache** mode, as explained below.
- The **ccf address** and **ecf address** parameters configured in the egress SIP interface.

The Net-Net SBC includes a P-Charging Function Address specific mode called **insert reg cache** which removes any existing charging headers and inserts previously saved value (including registration cache) or new values from the SIP interface configuration.

The **insert reg cache** mode's intended application is when the Net-Net SBC is on the edge of the IMS network. It should be configured on the access SIP interface while **delete and respond** mode is configured on the corresponding core-side SIP interface.

Missing SIP Interface

While you can explicitly configure the Net-Net SBC's charging header handling on the SIP interface where the message with charging headers is received, sometimes no SIP interface configuration exists where these messages are received. This can occur in IWF scenarios or when a response is sent for a local error (i.e. a 404 response). In such a case, charging header manipulations should still function. As a contingency, the Net-Net SBC can infer the incoming charging header mode from the outgoing SIP interface's configuration. The following are the mappings when the incoming SIP interface configuration is missing.

Outgoing mode	Inferred Incoming mode
None	None
Pass	Pass
Insert	Delete (PCFA), Delete and respond (PCV)
Delete	Insert
Insert reg cache	Delete and respond
Delete and respond	Insert reg cache (PCFA), insert (PCV)

RADIUS Accounting of Charging Headers

When the Net-Net 9200 creates either the P-Charging-Vector header or the P-Charging-Function-Address header, it inserts an entry in the RADIUS record to record the charging header data.

For a P-Charging-Vector header, the icid-value is saved to the P-Charging-Vector attribute in the radius record. If the Net-Net 9200 does not create a P-Charging-Vector header, but it receives a SIP message that already has the P-Charging-Vector header with an icid-value, the existing icid-value is written to the RADIUS record.

For a P-Charging-Function-Address header, the first CCF value is saved to the P-Charging-Function-Address attribute. When the Net-Net 9200 creates the P-Charging-Function-Address, the CCF value it inserts into the header is saved to the RADIUS record. If the Net-Net 9200 does not create a P-Charging-Function-Address header, but it receives a SIP message that already has the P-Charging-Function-Address with a CCF value, the existing CCF value is written to the RADIUS record.

Name	Value	Value Type
Acme-Session-Charging-Vector	54	string
Acme-Session-Charging-Function-Address	55	string

ACLI Configurations and Instructions

P-Charging-Vector header handling is enabled in the SIP interface configuration.

To configure P-Charging-Vector processing in a SIP interface:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

```
ACMEPACKET(session-router)#
```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-interface)#
```

4. Type **select** and the number of the pre-configured sip interface you want to configure.
5. **charging-vector-mode**—Set this parameter to the appropriate charging vector mode for your deployment. Valid values are: **none**, **pass**, **insert**, **delete**, **delete-and-respond**.
6. **operator-identifier**—Enter the inter operator identifier (IOI) string for this SIP interface. This string **MUST** begin with an alpha character
7. **charging-function-address-mode**—Set this parameter to the appropriate charging vector mode for your deployment. Valid values are: **none**, **pass**, **insert**, **delete**, **delete-and-respond**, **insert-reg-cache**.
8. **ccf-address**—Enter the CCF address value that will be inserted into the P-Charging-Function-Address header.
9. **ecf-address**—Enter the ECF address value that will be inserted into the P-Charging-Function-Address header.
10. Save and activate your configuration.

IP-PBX Surrogate Registration

The Net-Net SBC surrogate registration feature lets the Net-Net SBC explicitly register on behalf of a Internet Protocol Private Branch Exchange (IP-PBX). After you configure a surrogate agent, the Net-Net SBC periodically generates a REGISTER request and authenticates itself using a locally configured username and password, with the Net-Net SBC as the contact address. Surrogate registration also manages the routing of class from the IP-PBX to the core and from the core to the IP-PBX.

Integrating with IMS

With surrogate registration, the Net-Net SBC lets IP-PBXs integrate with the IP Multimedia Subsystem (IMS) architecture. The IP-PBX registers itself as if it were user equipment (UE), which triggers the implicit registration of all phone numbers associated with the IP-PBX.

Implicit registration means that the explicit registration of one address of record (AoR) triggers the implicit registration of all the other AoRs associated with that UE. The implicitly registered AoRs are passed back to the UE as P-Associated-URIs in the registration's 200 (OK).

IMS assumes that each SIP endpoint can register itself with its Serving-CSCF (S-CSCF). However, phones can be connected to SIP Integrated Access Devices (IADs) or SIP IP-PBXs. The Net-Net SBC performs SIP registration on behalf of the IP-PBX, IADs, or IMS core elements.

How it Works

The Net-Net SBC registers on behalf of the IP-PBXs and then stores the associated URIs returned by the Interrogating Call Session Control Function (I-CSCF). The calls from the phones behind the IP-PBX can be routed based on the cache entry the Net-Net SBC creates after it receives each phone's associated URI. Calls are routed using the service route, local policy or any other routing mechanism based on the associated session agent or session agent group. The Net-Net SBC also supports multiple registrations on behalf of a IP-PBX because the IP-PBX can support thousands of phones, but the registrar can only send 10 to 20 associated URIs in response to a single registration.

The Net-Net SBC replaces the Contact URI for requests from the IP-PBX to the core to match the registered value. The IMS core expects the inbound request's Contact URI to match a registered contact but requests from a PBX proxy will have the phones off the proxy and IP-enabled TDM PBXs might have the same Contact address but the user name might be the dialing phone number.

For calls from the IMS core to the IP-PBX, the Net-Net SBC replaces the Request-URI username with P-Called-Party-ID/To-URI username. The IMS cores sends INVITEs for the phones behind the IP-PBX with the registered Contact URI as the Request-URI instead of the AoR of the phones. The IP-PBX needs to see the phone's AoR in the Request-URI.

Registration

The Net-Net SBC uses the configuration information of the surrogate agent that corresponds to a specific IP-PBX. After the surrogate agents are loaded, the Net-Net SBC starts sending the REGISTER requests on their behalf. (You can configure how many requests are sent.)

If the Net-Net SBC receives 401 or 407 responses to REGISTER requests, it will use the Message Digest algorithm 5 (MD5) digest authentication to generate the authentication information. You need to specify the password. The Net-Net SBC also supports authentication challenge responses with the quality of protection code set to auth (qop=auth), by supporting the client nonce (cnonce) and nonce count parameters.

The Net-Net SBC creates a registration cache entry for each of the AoRs for which it is sending the REGISTER requests. When the Net-Net SBC receives the associated URIs, it checks whether the customer host parameter is configured. If it is configured, the Net-Net SBC changes the host in the received Associated-URI to the customer host. If it is not configured, the Net-Net SBC does not change the Associated-URI. It makes the registration cache entries that correspond to each of the Associated-URIs. The From header in the INVITE for calls coming from the IP-PBX should have one of the Associated-URIs (URI for a specific phone). If the Net-Net SBC receives a Service-Route in the 200 (OK) response, it stores that as well.

The Net-Net SBC uses the expire value configured for the REGISTER requests. When it receives a different expire value in the 200 OK response to the registration, it stores the value and continues sending the REGISTER requests once half the expiry time has elapsed.

REGISTER requests are routed to the registrar based on the configuration. The Net-Net SBC can use the local policy, registrar host and the SIP configuration's registrar port for routing.

If the Net-Net SBC is generating more than one register on behalf of the IP-PBX, the user part of the AoR is incremented by 1 and the register-contact-user parameter will also be incremented by 1. For example, if you configure the register-user

parameter as caller, the Net-Net SBC uses caller, caller1, caller2 and so on as the AoR user.

Routing Calls from the IMS Core

The calls coming from the core will have the Net-Net SBC's Contact-URI (which is sent in the REGISTER request) as the Request-URI. The Net-Net SBC looks for a registration entry that corresponds to this URI. After finding the registration entry and the corresponding surrogate agent, the Net-Net SBC looks for the routing mechanism it should use to route this INVITE to the IP-PBX. It uses the customer-next-hop configuration parameter to determine if it routes this call to the session agent, the session agent group, or directly to a particular IP address.

SIP

If the customer-next-hop parameter points to a SIP session agent or the SIP session agent group, the Net-Net creates a Route header using the session agent and modifies the Request-URI. It changes the user portion of the Request-URI to either the user portion of the P-Called-Party-ID header, if present, or to the user portion of the To header. The Net-Net SBC also changes the host portion of the Request-URI to the hostname configured in the customer-host configuration parameter. It makes the change because the domain name on the core side can be different than the domain name on the access IP-PBX side. The Net-Net SBC then uses the added Route header to properly route the call.

Note: Even though the **customer-next-hop** field allows specification of a SAG or FQDN, the functionality will only support these values if they resolve to a single IP address. Multiple IP addresses, via SAG, NAPTR, SRV, or DNS record lookup, are not allowed.

Registration Retry Mechanism

Although the surrogate agent uses the same retry mechanism used for SIP client transactions, it does not attempt additional retries if it receives a failure response until the entry expires. When the Net-Net SBC receives 401, 403, or 407 responses to requests that include authentication, the surrogate agent's automatic incrementing mechanism appends a number to the end of each registered username. Always starting at one, this number cannot appear in any other position in the username. The following two features are used for retrying registrations:

- **Registration retry**—You can configure the surrogate agent to retry registration after a failure, timeout, or transport error. You can set how many times the Net-Net SBC will attempt to register each user; a setting of zero means retries are unlimited. You can also define the number of seconds to wait before initiating a retry. The Net-Net SBC tracks each registration retry count and timers, and sends an SNMP trap when it reaches the maximum number of retries, which signifies failed registration.
- **Authentication loop prevention**—In order to mitigate authentication loops, which often are recurring password errors, the Net-Net SBC only allows permits the retransmission of one request. It now considers further 401, 403, or 407 responses to be errors and initiates the retry mechanism.

Automatic Increment Mechanism

The automatic increment feature works with the caret (^) in the register-user and register-contact-user fields. These carets define where the automatically generated incrementing number is inserted in the username. You can use multiple carets to define leading zeroes to insert; for example, the entry `user^^^^` will become `user0001`. You can also define the starting integer for the incrementing registrations.

For example, setting the AoR count to 20, the count start to 5, and using the value `user^^^^` for register-user and register-contact-user results in the incremented user registrations `user0005` through `user0025`. The carets can be placed at the beginning, middle, or end of a register user name.

Routing Calls from the IP-PBX

The Net-Net SBC looks for a match in the registration cache based on the From header or the P-Preferred-Identity header. The header should contain the user portion of one of the Associated-URIs that it received from the registrar in the 200 (OK) responses to REGISTER requests. It should also have the same hostname that is configured in the customer-host parameter. If that parameter is not configured, then the hostname should be the same as the one configured for the register-host parameter.

After the corresponding registration Service-Router entry is found, the Net-Net SBC looks at how to route this call. It uses the Service-Route for this endpoint to route the call, if it exists. If no Service-Route exists but the SIP interface's route-to-registrar parameter is enabled, the Net-Net SBC tries to route this to the registrar. You can configure the surrogate agent to override the SIP interface's route-to-register setting. If the surrogate agent's route-to-register parameter is set to disable, it takes precedence over the SIP interface's setting. The Net-Net SBC will not try to route the call to the registrar.

Configuring Surrogate Registration

You can configure surrogate registration using the ACLI. You need to configure a surrogate agent for each IP-PBX proxy for which the Net-Net SBC will be registering. Those parameters that are optional are marked, the rest are mandatory.

To configure the surrogate agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **surrogate-agent** and press <Enter>. The prompt changes to indicate you can configure individual parameters.


```
ACMEPACKET(session-router)# surrogate-agent
ACMEPACKET(surrogate-agent)#
```

From this point, you can configure surrogate agent parameters. To view all surrogate agent configuration parameters, enter a **?** at the system prompt.
4. **register-host**—Enter the registrar's hostname to be used in the Request-URI of the REGISTER request. This name is also used as the host portion of the AoR To and From headers.
5. **register-user**— Enter the user portion of the AoR.
6. **state**—Set the state of the surrogate agent to enabled (default) or disabled to indicate whether the surrogate agent is used by the application.
7. **realm-id**— Enter the name of realm where the surrogate agent resides (where the IP-PBX proxy resides). There is no default.
8. **description**— *Optional*. Enter a description of this surrogate agent.

9. **customer-host**—*Optional*. Enter the domain or IP address of the IP-PBX, which is used to determine whether it is different than the one used by the registrar.
10. **customer-next-hop**—Enter the next hop to this surrogate agent:
 - session agent group:
SAG: <session agent group name>
 - session agent:
<hostname> or <IPV4>
 - specific IP address:
<IPV4> or <IPV4: port>
11. **register-contact-host**—Enter the hostname to be used in the Contact-URI sent in the REGISTER request. This should always point to the Net-Net SBC. If specifying a IP address, use the egress interface's address. If there is a SIP NAT on the registrar's side, use the home address in the SIP NAT.
12. **register-contact-user**—Enter the user part of the Contact-URI that the Net-Net SBC generates.
13. **password**—If you are configuring the auth-user parameter, you need to enter the password used in case the registrar sends the 401 or 407 response to the REGISTER request.
14. **register-expires**—Enter the expires in seconds to be used in the REGISTER requests. Default value is 600,000 (1 week).
15. **replace-contact**—Set to enabled or disabled (default). This specifies whether the Net-Net SBC needs to replace the Contact in the requests coming from the surrogate agent. If this is enabled, Contact will be replaced with the Contact-URI the Net-Net SBC sent in the REGISTER request.
16. **route-to-registrar**—Set to enabled (default) or disabled. This indicates whether requests coming from the surrogate agent should be routed to the registrar if they are not explicitly addressed to the Net-Net SBC.
17. **aor-count**—Enter the number of registrations to do on behalf of this IP-PBX. If you enter a value greater than 1 (default), the Net-Net SBC increments the register-user and the register-contact-user values by that number. For example, if this count is 3 and register-user is john then users for three different register messages will be john, john1, john2. It does the same for the register-contact-user values.
18. **options**—*Optional*. Enter non-standard options or features.
19. **auth-user**—Enter the authentication user name you want to use for the surrogate agent. This name is used when the Net-Net SBC receives a 401 or 407 response to the REGISTER request and has to send the REGISTER request again with the Authorization or Proxy-Authorization header. The name you enter here is used in the Digest username parameter. If you do not enter a name, the Net-Net SBC uses the value of the register-user parameter.
20. **max-register-attempts**—Using a value from 0 (meaning registration attempts are unlimited) to 10, enter the number of times you want to attempt registration. The default for this parameter is 3.
21. **register-retry-time**—Enter the amount of time in seconds, between 30 and 3600 seconds, you want the Net-Net SBC to wait before reattempting registration. The default for this parameter is 300 seconds.

22. **count-start**—Change this parameter from its default of 1 if you want the automatic increment count to start at any other number when the Net-Net SBC performs multiple registrations. The valid range is 0 through 999999999.
23. Save and activate your configuration.

Viewing Surrogate Agent Statistics

You can view surrogate agent statistics using the ACLI show command.

show sip cache

You can use the **show sip cache** command to display the following surrogate agent statistics:

- Surrogate registration cache entries.
- Total number of surrogate registers sent
- Total number of surrogate register timeouts
- Total number of surrogate register rejects.

show sip endpoint

You can use the **show sip endpoint** command to view the UA-contact, Contact-URI, and expire time for this surrogate registration.

SNMP Support

Each time a user fails in attempts to register more times than you have allowed in your configuration (in the **max-register-attempts** parameter), the Net-Net SBC sends the new `apSysMgmtSurrogateRegFailed` traps, with associated varbinds for the failing AoR and registrar-host. This has been added to `ap-sysmgmt.mib`.

```

apSysMgmtSurrogateRegHost OBJECT-TYPE
SYNTAX      DisplayString (SIZE (1..255))
MAX-ACCESS  accessible-for-notify
STATUS      current
DESCRIPTION
"The registrar host name for a surrogate registration"
::= { apSysMgmtMonitorObjects 35 }

apSysMgmtSurrogateRegAor OBJECT-TYPE
SYNTAX      DisplayString (SIZE (1..255))
MAX-ACCESS  accessible-for-notify
STATUS      current
DESCRIPTION
"The address of record used in a surrogate registration"
::= { apSysMgmtMonitorObjects 36 }

apSysMgmtSurrogateRegFailed NOTIFICATION-TYPE
OBJECTS     { apSysMgmtSurrogateRegHost,
              apSysMgmtSurrogateRegAor }
STATUS      current
DESCRIPTION
" The trap will be generated if a surrogate registration
  failed after the maximum configured attempts."
::= { apSystemManagementMonitors 39 }

```

Wildcarded PUID in P-Associated-URI

In the IMS core, a PBX often registers its own base telephone number with the S-CSCF, and the S-CSCF obtains a wildcarded public user identity, represented by a regular expression that describes all extension numbers associated with the registering PBX.

The S-CSCF includes the wildcarded Public User Identifier (PUI) in the 200OK response to the REGISTER message that the Net-Net SBC sent it, prompted by the initial registering endpoint. Upon receiving the 200OK, the Net-Net SBC uses the wildcarded puid to implicitly treat subsequent INVITEs from PBXs extensions that match the wildcard as registered endpoints.

For all calls directed at telephone numbers that are implicitly registered by the wildcarded puid regular expression, the Net-Net SBC terminates them the servicing PBX.

This behavior is configured by adding the **wildcard-puid-match** option on the access side SIP interface when requests originate there.

Wildcarded Public User Identity

A wildcarded PUI consists of a delimited regular expression located either in the userinfo portion of the SIP URI or in the telephone-subscriber portion of the Tel URI. The regular expression in the wildcarded PSI shall take the form of Extended Regular Expressions (ERE) as defined in chapter 9 in IEEE 1003.1-2004 Part 1. The delimiter is the exclamation mark character ("!").

Using the following PUI as an example:

```
sip:chatlist!.*!@example.com
```

The following PUIs will match:

```
sip:chatlist1@example.com
sip:chatlist2@example.com
sip:chatlist42@example.com
sip:chatlistAbC@example.com
sip:chatlist!1@example.com
```

ACLI Instructions

To configure wildcarded PUID registration:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI **select** command) the SIP interface that you want to edit.

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **wildcard-puid-match**, and then press <Enter>.

```
ACMEPACKET(sip-interface)# options +wildcard-puid-match
```

If you type **options wildcard-puid-matches**, you will overwrite any previously configured options. In order to append the new option to the **sip-interface**'s options list, you must prepend the new option with a "plus" sign as shown in the previous example.

5. Save and activate your configuration.

Implicit Service Route

In an IMS architecture, the Service-Route header indicates the network's requirement for where to direct all subsequent signaling from a given UE. Upon receipt of the Service-Route header(s) in a 200 OK response to a REGISTER request, the Net-Net 9200 stores the values contained therein and preloads them when issuing new requests (such as an INVITE or a SUBSCRIBE) from the associated endpoint.

Although not all SIP networks fully support RFC 3608, some still require UEs to send all call traffic to the network element(s) with which they originally registered. Further, some P-CSCFs do not pass back the Service-Route headers in response to the UE's REGISTER request. This situation requires that a logical connection is made between the UE and the network element to which it registers.

In these scenarios, the Net-Net 9200 may create an implicit Service-Route for the success (2xx class) response it receives from the P-CSCF. An implicit Service-Route is useful for creating a binding between a UE and a next-hop proxy, even when the Net-Net 9200 can route to many potential, yet logically equivalent next-hops.



The Net-Net 9200's implicit Service-Route feature is enabled with the implicit-service-route parameter. It is configured on the access-side SIP interface where a UE registers.

ACLI Instructions and Examples

To enable implicit Service-Route support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **select** and the number of the pre-configured sip interface you want to configure.


```
ACMEPACKET(sip-interface)# select 1
```

5. **implicit-service-route enabled**—Set this parameter to `enabled` to turn on the implicit Service-Route feature. The default is disabled.

```
ACMEPACKET(sip-interface)# implicit-service-route enabled
```

6. Save and activate your configuration.

P-CSCF: P-Asserted Identity in Responses

The Net-Net SBC:

- Caches a copy of the P-Called-Party-ID header when it receives one of the following destined for a UE prior to forwarding the request:
 - An initial request for dialog
 - A request for a standalone transaction
 - A request for an unknown method that does not related to an existing dialog

The SIP interface receiving the request should have the SIP IMS feature enabled.

- Removes the P-Preferred-Identity header (if present) and inserts a P-Asserted-Identity header with the value saved from the P-Called-Party-ID header on receipt of a 1xx or 2xx response.

Important Notes

Note the following:

- The endpoint to which the response is being sent must be a trusted endpoint. The option **disable-ppi-to-pai** should not be configured in the global SIP configuration's **options** list.
- If the P-Preferred-Identity header is present in the response, the Net-Net SBC will delete the header.
- If the P-Asserted-Identity header is present in the response, the Net-Net SBC will overwrite that -Asserted-Identity.

ACLI Instructions and Examples

This behavior is enabled automatically. You do not need to perform any configuration steps.

P-CSCF Redundancy with Implicit Service Route

You can configure multiple P-CSCFs in a core network to provide redundant P-CSCF functionality. Configuring multiple P-CSCFs is performed similarly to configuring any type of redundant session agent (SA), either through DNS for a single SA or with a session agent group (SAG). If an SA (or member of an SAG) goes out of service, and the IP address is unreachable, the Net-Net 9200 uses the next IP address for the SA (or member of an SAG). Also the Net-Net 9200 can load balance P-CSCFs by using a load balancing SAG strategy such as round-robin.

Tracking UE per P-CSCF

To ensure continuity of service, the Net-Net 9200 tracks the relationship between a UE and the P-CSCF it registers to. When a P-CSCF goes out of service and fails over to the next P-CSCF (as an SA or member of an SAG), the Net-Net 9200 notes the time of the failover event. When a UE associated with the failed P-CSCF re-registers, the Net-Net 9200 sees that the previous P-CSCF has experienced a failover. Upon a

P-CSCF failover, the Net-Net 9200 redirects the associated UE's traffic to the new P-CSCF as the registrar/proxy.

To configure the Net-Net 9200 to direct a UE to use a P-CSCF that has been placed in service after the previous P-CSCF failed, use the parameter that enables the implicit service-route feature as described in the previous section.

Registration Invalidation

The Net-Net 9200 includes provisions to explicitly invalidate current registrations in certain instances. If the P-CSCF (configured as a SA) goes out of service, or if the active IP address that specifies a P-CSCF does not point to the active P-CSCF, the Net-Net 9200 can invalidate the current endpoint registrations to that SA. The new configuration, when enabled, forces the Net-Net 9200 to act as if the registrations have expired, thus forcing the endpoints to re-register. The Net-Net 9200 responds back to the registering endpoint with a 403 response (if the allow-anonymous parameter in the sip-ports configuration element is set to registered).

ACLI Instructions and Examples

To enable registration invalidation support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. Type **select** and the number of the pre-configured sip interface you want to configure.


```
ACMEPACKET(session-agent)# select 1
```
5. Type **invalidate-registrations enabled** and press <Enter>. This completes enabling the implicit Service-Route feature.


```
ACMEPACKET(session-agent)# invalidate-registrations enabled
```
6. Save and activate your configuration.

Contact Header Contact Formatting

Some P-CSCF implementations require an extra parameter in the Contact header of the register message for full compatibility. This parameter should contain the public IP address of the UE.

The `contact-firewall` option is configured on the SIP interface attached to the core network.

When entering the `contact-firewall` option without a value, the 9200 will add a parameter to the Contact header called `firewall`. The value will be `IPaddress:port`.

Note the appearance of the contact header in the following example and how it includes the endpoint's public IP address.



To change the parameter name from "firewall" to a name of your choosing, enter the option as `contact-firewall="parameter_name"`.

ACLI Instructions and Examples

To add a UE's public IP address and port to the contact header:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **select** and the number of the pre-configured sip interface you want to configure.


```
ACMEPACKET(sip-interface)# select 1
```
5. Type **option +contact-firewall** and press <Enter>. This appends the new contact-firewall option to the existing list of configured SIP interface options.


```
ACMEPACKET(sip-interface)# option +contact-firewall
```
6. Save and activate your configuration.

SIP IMS and Privacy

In order to determine the trust level on the ingress side correctly, you need to set the **ppi-to-pai** parameter in the SIP interface configuration. If the sip-ims-feature parameter is enabled on the incoming SIP interface, then the Net-Net 9200 applies the IMS rules for determining PAI. And if the PAI is based on the IMS processing, the PPI will be removed. Otherwise, the Net-Net 9200 removes the PPI unless you:

- Set the ingress SIP interface's **ppi-to-pai** parameter to the replace mode
- Set the ingress SIP interface is designated as trusted and **ppi-to-pai** is also set to trusted

The egress processing of this parameter is applied if the PPI is not removed.

On the egress SIP interface, the Net-Net 9200 checks the ppi-to-pai parameter only if the message's destination is designated trusted. If the destination is untrusted, then the Net-Net 9200 does not reference this parameter.

There are four valid values for the **ppi-to-pai** parameter:

- **none**—The Net-Net 9200 performs no conversion
- **trusted**—The Net-Net 9200 checks the configured trust level for the ingress endpoint; if it is untrusted, then PPI headers in the message are deleted; if it is trusted, then PPIs are converted to PAIs
- **remove**—The Net-Net 9200 removes PPI headers
- **always**—The Net-Net 9200 always converts PPIs to PAIs

ACLI Instructions and Examples

To set the PPI to PAI conversion mode:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **ppi-to-pai**—Enter the conversion mode that you want to use; valid values appear in the description above.

Emergency Session Handling

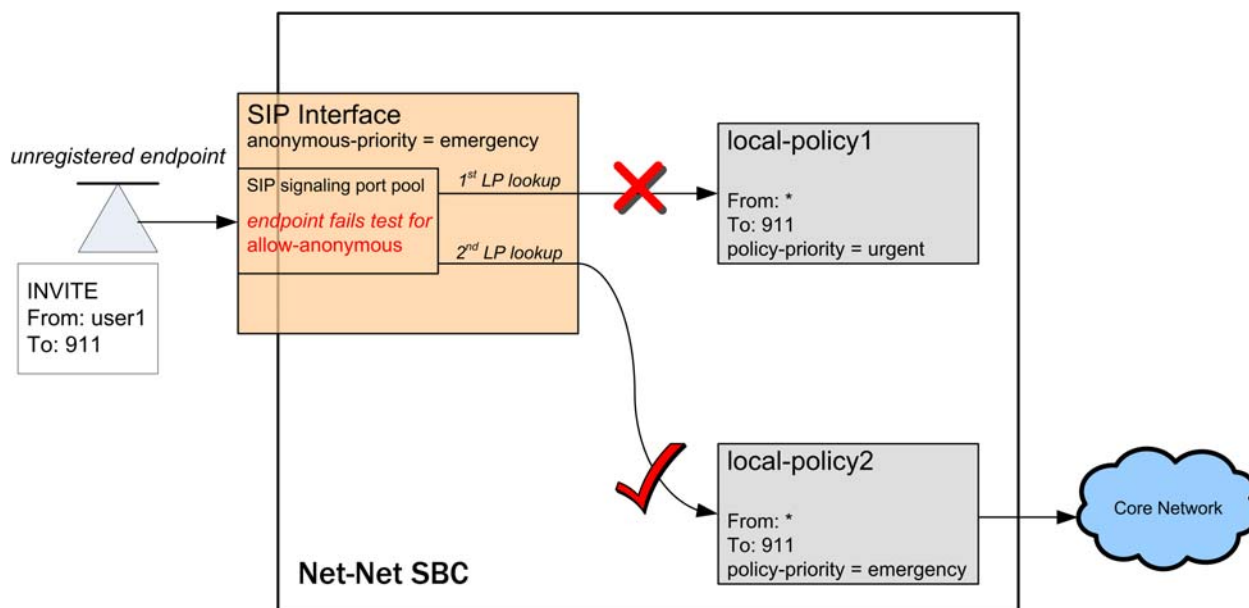
The Net-Net SBC provides a mechanism to handle emergency sessions from non-allowed endpoints. An endpoint is designated as non-allowed if it fails the admission control criteria specified by the allow-anonymous parameter in the SIP Ports configuration element.

When the Net-Net SBC receives a non-allowed emergency request, it performs a local policy lookup for a matching local policy. An emergency local policy could be configured to match if the To: header in a SIP message was addressed to 911.

An emergency policy priority selection criteria has been added to both the SIP interface and the local policy configuration elements. In the SIP interface, the parameter is called anonymous-priority. In the local policy, the parameter is called policy-priority.

For the Net-Net SBC to choose a local policy to route an emergency call, the emergency policy priority value on the local policy must be equal to or greater than the emergency policy priority value on the SIP interface where the emergency message was received. In this scheme, an emergency policy priority value of none is the lowest value and an emergency policy priority value of emergency is the highest.

When a match is made between all existing local policy criteria and the emergency policy priority, the emergency call will be sent to the core network according to the chosen local policy. In addition, the policy priority value of the chosen local policy is inserted into the Priority header of the core-bound SIP message..



Emergency Session Handling Configuration Procedures

Note the value of the allow-anonymous parameter in the SIP interface's SIP Ports for the incoming interface you are configuring. When an incoming emergency call from an unregistered endpoint can not be characterized by this setting, the Net-Net SBC will use the following means to route the call.

Set the anonymous-priority parameter in the incoming SIP interface. This parameter specifies that for an INVITE received from an anonymous endpoint, the Net-Net SBC will choose a local policy of equal or greater policy priority for outbound routing.

Next, set the policy-priority parameter located in the local-policy configuration element. Most likely, this local policy will route messages to SIP devices that act on emergency calls. The local policy is selected when its value (or above) matches the anonymous-priority parameter in the sip-interface that receives the incoming phone call from an unregistered endpoint.

The enumerated values for both the anonymous-priority and policy-priority are: none, normal, non-urgent, urgent, emergency.

ACLI Instructions and Examples

To set the anonymous priority for a message received in a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. Type **select** and the number of the SIP interface you want to configure.


```
ACMEPACKET(sip-interface)# select 1
```
5. **anonymous-priority**—Set the policy priority for this SIP interface. The default is **none**. The valid values are:
 - none | normal | non-urgent | urgent | emergency
 This completes the configuration.


```
ACMEPACKET(sip-interface)# anonymous-priority emergency
```
6. Save your work using the ACLI **done** command.

Setting Policy Priority

To set the policy priority for a local policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. Type **select** and the number of the local policy you want to configure.


```
ACMEPACKET(local-policy)# select 1
```
5. **policy-priority**—Enter the policy priority for this local policy. The default is **none**. The valid values are:
 - none | normal | non-urgent | urgent | emergency

This completes the configuration.

```
ACMEPACKET(local-policy)# policy-priority emergency
```

6. Save your work using the ACLI **done** command.

Recurse on 305 Only

When the Net-Net SBC is deployed as a P-CSCF in an IMS network, 3GPP TS 24.299 dictates that it must recurse only on 305 (Use Proxy) responses.

The Net-Net SBC's normal behavior, when the redirect-action set to proxy, is to not recurse when it receives a 3xx messages. The Net-Net SBC sends the 3xx SIP response back to the sender. In addition, the request URI is changed to contain the contact value in this response message.

You can enable the Net-Net SBC to recurse on a 305 (Use proxy) message. Upon receiving all other 3xx messages, the Net-Net SBC will not recurse and will send a 3xx SIP response back to the sender.

This behavior is configured by setting the **redirect action** to **recurse 305** in either the session agent or SIP interface configuration elements.

When recurse on 305 is configured, the request URI in the recursed reply will not change from the one in the original request. The Route header in the recursed reply will include URI contained in the Contact header of the 305 Use Proxy reply.

To configure a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **redirect-action**—Set this parameter to **recurse-305**.

Save and activate your configuration.

DTMF Interworking

Multimedia devices and applications can exchange user-input DTMF information end-to-end over IP networks. The Net-Net 9200 provides the capabilities required to interconnect networks and devices that use different DTMF indication signaling protocols.

DTMF Indication

There are three ways to convey DTMF information for packet-based communications:

- DTMF audio tones: DTMF digit waveforms are encoded inline with voice packets. This method only works with uncompressed audio codecs like G.711. Compressed audio codecs like G.729 and G.723 are incompatible with DTMF audio. DTMF audio is also referred to as in-band tones.
- Out-of-band signaling events:
 - H.245 defines out-of-band signaling events (UII) for transmitting DTMF information. The H.245 signal or H.245 alphanumeric methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of through the RTP channel. The tones are transported in H.245 UII messages.

All H.323 version 2 compliant systems are required to support the H.245 alphanumeric method, while support of the H.245 signal method is optional.
 - SIP INFO messages with Content-Type: application/dtmf-relay define out-of-band signaling events for transmitting DTMF information. SIP INFO messages separate DTMF digits from the voice stream and send them in their own signaling message.
- RTP named telephony events (NTE): RFC 2833 telephone-events are a standard that describes how to transport DTMF tones in RTP packets according to section 3 of RFC 2833. Of the three RTP payload formats available, the Net-Net 9200 supports RTP NTE.

RFC 2833 telephone-event

RFC 2833 specifies a way of encoding DTMF-indications in RTP media streams. It does not encode the audio of the tone itself, instead data represents the sent tone. RFC 2833 can be used with both SIP and H.323.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for RFC 2833 packets using a SDP or H.245 Terminal Capability Set (TCS), which tells the other side what payload type number to use

when sending the named event packets to the local device. Most devices use payload type number 101 for RFC 2833 packets, although no default is specified in the standard.

The RFC 2833 packet's RTP header also makes use of the timestamp field. Because events often last longer than the RFC 2833 packets sending interval, the timestamp of the first 2833 packet for an event represents the beginning reference time for subsequent RFC 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

See RFC 2833 and draft-ietf-avt-rfc2833bis-07.txt for more information.

SIP INFO Messages

SIP INFO messages can send indications of DTMF audio tones between peers as part of the signaling path of the call. Upon receipt of a SIP INFO message with DTMF content, the gateway generates the specified DTMF tone on the receiving end of the call.

H.323 UII Messages

H.323 UII messages can send indications of DTMF audio tones between peers as part of the signaling path of the call.

DTMF Transfer Processing Overview

Enabling 2 UAs to communicate different DTMF indications with each other is facilitated in two steps. First, the Net-Net 9200 takes an active role in the SDP negotiation between two UAs, this is the capability negotiation step. You can configure your system to suggest the DTMF indication methods that each endpoint can and can not use.

The second step is translation evaluation. After the SDP negotiation is complete, based on system configuration and the media support on each call leg, the Net-Net 9200 performs DTMF indication conversion or passive forwarding of DTMF indication messages between UAs.

Capability Negotiation

SDP capability negotiation is the first phase of enabling DTMF transfer. The completion of the SDP offer/answer exchange yields a set of supported codecs between each UA and the Net-Net 9200.

For DTMF transfer consideration, SDP manipulation is directed by parameters set in one of three configuration elements:

1. transcoding policy
2. signaling interface's RFC 2833 mode
3. session agent's RFC 2833 mode

The Net-Net 9200 performs SDP manipulation (addition, removal, or modification of supported codecs) toward the called party first by any applicable transcoding policies. If one or more of the actions which define [telephone-event Modification by Transcoding Policy \(549\)](#) occurs, then SDP manipulation is only performed by the transcoding policy configuration, not by RFC 2833 mode parameters in the signaling interface or session agent.

If a transcoding policy attached to either the ingress or egress realm triggers SDP manipulation, then the other realm uses transcoding policy for any telephone-event

SDP manipulation; none of the RFC 2833 Mode configurations are used for SDP manipulation.

Note: If the call does not trigger the evaluation of any transcoding policies, all DTMF transfer processing is only subject to RFC 2833 Mode rules.

If none of the [telephone-event Modification by Transcoding Policy \(549\)](#) occur, then the Net-Net 9200 performs SDP manipulation according to the RFC 2833 Mode parameters in the signaling interface or session agent.

SDP Manipulated by Transcoding Policy

When a call is received, any applicable transcoding policies are applied and evaluated. If telephone-event SDP is modified by transcoding policies, then SDP manipulation by RFC 2833 Mode is not performed for either side of the call.

telephone-event Modification by Transcoding Policy

For qualifying if telephone-event modification in SDP was performed by transcoding policy, one of the following events had to have happened:

- Transcoding policy explicitly deleted telephone-event by configuring **allow telephone-event:no**
- Transcoding policy explicitly added telephone-event by configuring **add-on-egress telephone-event**
- Transcoding policy implicitly denied telephone-event by allowing one or more codecs but not adding telephone-event to the allow list
- Transcoding policy has **audio:no** configured in the allow list

The following three cases highlight how transcoding policies can manipulate telephone-event SDP. Once any of these cases occurs, SDP manipulation by RFC 2833 Mode parameter will not be performed. For a full explanation of transcoding policy configuration, see the Transcoding chapter.

1. telephone-event added to SDP: The transcoding policy adds telephone-event to the SDP sent to the egress realm. The UA supports telephone-event.



2. telephone-event maintained in SDP: The transcoding policy maintains the offered telephone-event in the SDP sent into the egress realm. Although

telephone-event was not answered in the egress realm, telephone-event is added back to the offerer-side SDP because of the add-on-egress setting in the ingress realm.



3. telephone-event removed from SDP: transcoding policy removes telephone-event from the initial SDP offer. telephone-event is not forwarded to the Answerer, and subsequently not returned from the answerer, or forwarded again to the offerer.



SDP Manipulated by RFC 2833 Mode

If none of the [telephone-event Modification by Transcoding Policy \(549\)](#) events occur, as previously explained, SDP may still be modified by RFC 2833 Mode parameter if preferred or dual mode is configured.

The RFC 2833 mode parameter functions similarly to the add-on-egress parameter; it suggests telephone-event support, by adding it in SDP if not already there. This parameter must be set to preferred or dual to add telephone-event to SDP.

Transparent RFC 2833 Support

Setting a signaling interface or session agent's RFC 2833 mode to transparent disables the addition of RFC 2833 telephone-event to SDP upon egress. The Net-Net 9200 passes the offered SDP capabilities to the next-hop signaling element.

Preferred RFC 2833 Support

Setting a signaling interface or session agent's RFC 2833 mode to preferred indicates that the RFC 2833 telephone-event DTMF transfer method is the preferred method for sending a DTMF indication. In the capability negotiation phase a telephone-event media type will be inserted in the outgoing SDP offer, if it was not present in the original offer.

1. In the following example RFC 2833 mode is set to preferred on the egress side of the call. Because there is no telephone-event in the SDP, and RFC 2833 mode is set to preferred, the Net-Net 9200 adds telephone-event to the SDP offer.



2. In the following example, RFC 2833 is set to preferred mode on the SDP offer side of the call. The Net-Net 9200 maintains the telephone-event support even though telephone-event is not supported on the SDP answerer's side of the call.



RFC 2833 Payload Type Mapping

The Net-Net 9200 does not require that call legs use the same media type for telephone-event. If each call leg uses a different media type value, the Net-Net 9200 facilitates payload type mapping to ensure the telephone-event media stream be reliably transported across the call.

- On the SDP offer side, when the Net-Net 9200 returns its SDP answer, it uses the same media type that the SDP offerer offered.
- The Net-Net 9200 forwards the originally offered telephone-event media type to the SDP answerer. If telephone-event was added by RFC 2833 mode, the Net-Net 9200 adds telephone-event with the media type value configured in the RFC 2833 payload parameter. If telephone-event was added by a transcoding policy, the Net-Net 9200 adds telephone-event with the media type value configured in the media profile.
- If the SDP answerer returns a new value for telephone-event, the Net-Net 9200 still supports RFC 2833 on that side of the call and uses the media type that the answerer sent.

Translation Evaluation

After SDP has been negotiated, the Net-Net 9200 determines what types of DTMF translation takes place for the call. The Net-Net 9200 sequentially evaluates the following rules for each call leg to determine what DTMF indication type it will forward to an endpoint.

1. RFC 2833—When the SDP offer/answer exchange resolves to both the Net-Net 9200 and the endpoint supporting RFC 2833 on one side of the call, the Net-Net 9200 will send DTMF indications in RFC 2833 format.
2. DTMF audio tones—Three conditions must be met for the Net-Net 9200 to support DTMF audio tones, as transcoded from another DTMF indication form:
 - The applicable transcoding policy's dtmf in audio parameter is set to preferred
 - The endpoint and Net-Net 9200 have negotiated to a DTMFable codec (G711)
 - Transcoding resources are available

Note: Because of rule number one, rule number two can not happen if RFC 2833 is supported in SDP—Only one media-based DTMF transfer method, RFC 2833 or DTMF audio tones may be used on a call leg.

3. If neither RFC 2833 nor DTMF Audio tones are supported on a call leg, as a result of SDP negotiation, then the Net-Net 9200 forwards DTMF indication messages to that side in signaling message format (SIP INFO or H.323 UII).

In the following images that illustrate DTMF transfer scenarios, a gears icon appears when relevant. This icon indicates that the Net-Net 9200 performs DTMF indication processing, creating DTMF audio tones or RFC 2833 telephone-event messages from another form of DTMF indication.

RFC 2833 Sent by Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in RFC 2833 format. The SDP answerer can receive DTMF indications in the format identified in each example.

RFC 2833 to RFC 2833

When the SDP offer and answer sides of a call both support RFC 2833, the Net-Net 9200 forwards RFC 2833 messages between both sides of the call. No processing is used to transform these DTMF-indication messages to another format.

Note: When the audio stream is transcoded, DTMF audio is completely removed from the audio stream.



A SIP INFO/H.323 UII message received from either the offerer or answerer is forwarded unconverted to the other side of the call.

Note: If there is no audio transcoding enabled for this call, and the egress side is set to dual, a received SIP INFO/H.323 UII message will not be converted to both RFC 2833 and SIP INFO/H.323 UII messaged for sending to the other side of the call.

If DTMF audio tones are received from either the offerer or answer, they are forwarded unconverted to the other side (when the audio portion of the call is not transcoded).

RFC 2833 to DTMF Audio Tones

When the SDP offer side supports RFC 2833, and the SDP answer side supports the three DTMF Audio Tone conditions and does not support RFC 2833, the Net-Net 9200 converts from RFC 2833 to DTMF audio tones for the call.



A SIP INFO/H.323 UII message received by the Net-Net 9200 from either the offerer or answerer is converted into the DTMF transfer method that the previous diagram

shows for the egress side of the message. In this case, transcoding resources are used.

RFC 2833 to SIP INFO / H.323 UII

When the SDP offer side supports RFC 2833 and the SDP answer side does not support the DTMF conditions nor RFC 2833, the Net-Net 9200 converts from RFC 2833 to SIP INFO / H.323 UII.



If a SIP message is received from the offerer, it is forwarded unconverted to the answerer.

DTMF Audio Tones Sent by Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in DTMF audio tones format. The SDP answerer can receive DTMF indications in the format identified in each example.

DTMF Audio to DTMF Audio

If the SDP offer and answer sides both support the same type of G711 codec, the audio stream is forwarded between the two sides without processing.



If the two sides of the call support DTMF audio tones, but use different audio codecs, and the SDP answer side supports the three DTMF Audio Tone conditions and does not support RFC 2833 then the Net-Net 9200 will preserve DTMF audio tone indication across the call.

Transcoding resources are used only if different audio codecs are used or the [Override Preferred DTMF Audio \(560\)](#) feature is enabled.

DTMF Audio to RFC 2833

When the SDP offer side supports DTMF audio tones, and the SDP answer side supports RFC 2833, and transcoding resources are available, and does NOT support either or both of the first two DTMF Audio tone conditions, then the Net-Net 9200 will convert incoming DTMF audio tones to outgoing RFC 2833 packets.



Transcoding resources are always required in this scenario.

DTMF Audio to SIP

When the SDP offer side supports DTMF audio tones, and the SDP answer side does not support RFC 2833, and does not support the three DTMF Audio Tone conditions, then the Net-Net 9200 converts incoming DTMF audio tones to SIP INFO / H.323 UII messages.



Transcoding resources are always required in this scenario.

SIP INFO / H.323 UII Sent By Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in SIP INFO / H.323 UII message format. The SDP answerer can receive DTMF indications in the format identified in each example.

SIP INFO / H.323 UII to RFC 2833

When the SDP offer side sends a SIP INFO / H.323 UII message, and the SDP answer side supports RFC 2833, then the Net-Net 9200 will convert incoming SIP INFO / H.323 UII messages to outgoing RFC 2833 packets.



SIP INFO / H.323 UII to DTMF Audio

SIP INFO / H.323 UII will only be converted to DTMF audio tones only if RFC 2833 is not supported, dtmf-in-audio is enabled, the answer side supports a G711 codec, and transcoding resources are available.



Transcoding resources are always required in this scenario.

SIP INFO to SIP INFO

When the SDP offer side sends a SIP INFO / H.323 UII message and the SDP answer side does not support RFC 2833 and does not support the three DTMF audio tone

conditions, the SIP INFO / H.323 UII message will always be forwarded as the like SIP INFO / H.323 UII message.



SIP INFO to H.323 UII

When the two call legs communicate with the Net-Net 9200 with a different signaling protocol, i.e., SIP and H.323, the Net-Net 9200 seamlessly enables signaling interworking among these native DTMF indications. This requires no additional configuration.

When one side sends a SIP INFO / H.323 UII message and the SDP answer side does not support RFC 2833, does not support the three DTMF audio tone conditions, and uses a different signaling protocol, SIP INFO will be converted to H.323 UII and vice versa.



Dual Mode

Dual mode is used to send both RFC 2833 and the protocol-specific DTMF indication to a UA when possible: H.245 UII from an H.323 stack and SIP INFO from a SIP interface.

To consider dual mode scenarios, the Net-Net 9200 sets up the call SDP as explained in the [Capability Negotiation \(548\)](#) section. At the conclusion of the capability negotiation, the Net-Net 9200 is configured to support DTMF Audio tones or RFC 2833 independently for each side of the call.

When the call leg supports RFC 2833 as the means of DTMF transfer, the Net-Net 9200 checks if the SIP interface's (or session agent's) RFC 2833-mode parameter is

configured to dual. If it is, the Net-Net 9200 sends both RFC 2833 and SIP INFO / H.323 UII messages to the UA on this side of the call.

Note: Whether RFC 2833 support was initiated between the Net-Net 9200 and the UA by a transcoding policy or by the `rfc2833-mode` parameter, the Net-Net 9200 looks to the `rfc-2833` parameter to consider if dual mode is supported.

When the call leg supports DTMF audio tones as the means of DTMF transfer, the Net-Net 9200 checks if the transcoding policy's `dtmf-in-audio` parameter is configured to dual. If it is, the Net-Net 9200 sends both DTMF audio tones and SIP INFO / H.323 UII messages to the UA on this side of the call.

P-Dual-Info Header

When the Net-Net 9200 forward both media DTMF indication and signaling based DTMF indication for the same received DTMF indication, a P-Dual-Info header is added to the forwarded signaling message. You can configure the appearance of the header with the **dual-info** option. The default header appearance is:

```
P-dual-info: true
```

P-Dual-Info headers are only inserted into SIP INFO messages.

Example 1

In this example, RFC 2833 is supported on the egress side of the call. The egress SIP interface or session agent's `rfc2833-mode` is set to dual mode. When the Net-Net 9200 forwards RFC 2833 to the SDP answerer, it also creates and forwards a corresponding SIP INFO / H.323 UII message toward the target.



Example 2

In this example RFC 2833 telephone-event is not supported on the egress side of the call, but DTMF audio tones are. If the Net-Net 9200 receives an RFC 2833 message, it is converted to DTMF audio tones. When the Net-Net 9200 forwards DTMF audio

tones to the SDP answerer, it also creates and forwards a corresponding SIP INFO / H.323 UII message toward the target.



Identical Inband with Signaling DTMF Transfer Exception

Endpoints may send signaling-based DTMF indication (SIP INFO or H.323 UII) to the Net-Net 9200 at any time. Most of the time they are passed through the Net-Net 9200 unchanged. You can enable an exception to this behavior forcing signaling messages to either RFC 2833 or DTMF audio tones depending on the call's DTMF transfer mode. The two parameters to enable their respective exceptions are located in the media manager configuration.

Note: These behavior exceptions are only applicable when both sides of the call support and prefer the same DTMF transfer mode.

Override Preferred RFC 2833

When RFC 2833 is supported on both sides of the call, it is the preferred method of DTMF indication transport. To override this behavior, enable the `translate non rfc 2833` parameter.

**Override Preferred DTMF Audio**

When DTMF audio tones are supported on both sides of the call, it is the preferred method of DTMF indication transport. To override this behavior, enable the `translate non inband event` parameter.



Note that enabling one or both of these exceptions can cost DTMF translation resources whether they occur in the network processors or on TCUs. The `translate-non-inband-event` exception is especially costing because it reserves transcoding for all calls that resolve to support for DTMFable to DTMFable codecs.

H.323 DTMF Transfer Limitation

If a call leg is operating in H.323 fast start mode, and its `rfc 2833` mode is set to transparent, the Net-Net 9200 can not support telephone-event. Therefore, if the UA on this call leg asserted that it can support telephone-event in its TCS message,

regardless of telephone-event support on the other call leg, the Net-Net 9200 will respond that telephone-event is not supported.

If you want telephone-event supported on a Fast Start call leg, its RFC 2833 mode must be set to preferred or dual.

DTMF Transfer for Spiral Calls

A spiral call occurs when a call's signaling messages loop back through the Net-Net 9200. Most commonly the signaling path is from one UA, through the Net-net 9200, to a call server, back through the Net-Net 9200, to another UA. The media path is from one UA, through the Net-Net 9200, to the other UA. For DTMF indication processing, only the call legs between the endpoints and Net-Net 9200 are considered.

The Net-Net 9200 evaluates that the signaling path to and from the call server terminates on the same IP address and port on the Net-Net 9200. This means that it's a spiral call. In addition, the media IP addresses and ports in the SDP indicate that the Net-Net 9200 does not need to send the media into the call server's realm.

An issue occurs when DTMF indication is relayed either by RFC 2833 or DTMF audio tones for a spiral call. Since the DTMF indication is in the media path, the call server remains unaware of the signaling; no media-based DTMF indication digits will ever reach the call server. In order to include the call server in the DTMF-indication signaling, you should set the dtmf-in-audio and/or rfc-2833 mode used for the realm or signaling interface where the call server is to dual.

P-Dual-Info Header

The P-Dual-Info header is used in a spiral call scenario, when the call leg to (and from) the call server is set to dual. While the media portion of a spiral call goes from endpoint to endpoint through the Net-Net 9200, the signaling portion of the call loops through a call server.

The Net-Net 9200 inserts a P-Dual-Info header into a SIP info message sent to the call server, which in turn forwards the SIP INFO message back to the Net-Net 9200. Seeing the P-Dual-Info header, the Net-Net 9200 knows not to forward this SIP INFO message to the target endpoint because it would be a duplication of DTMF indication already sent to the endpoint in media format.



DTMF Transfer Hardware Processing

DTMF transfer processing, the conversion between two DTMF transfer types, occurs in either the NPU's network processors (NPs) or the TCU's transcoding modules (TCMs). Understanding where the processing takes place is important to determine which subsystem uses extra processing load per conversion.

There are a few rules you can use to determine which subsystem performs the DTMF transfer processing:

- If audio transcoding is enabled for the call, DTMF transfer processing occurs in the TCMs.
- If DTMF audio tones are generated from RFC 2833 or from signaling messages (SIP INFO or H.323 UII), DTMF transfer processing occurs in the TCMs.
- If the global translate non inband event parameter is enabled, DTMF transfer processing occurs in the TCMs.
- If signaling to RFC 2833 processing occurs in either direction of the call, and the previous 3 conditions are not valid, DTMF transfer processing occurs in the NPs.

DTMF Transfer Configuration

RFC 2833 Session Agent Configuration

Session agents, used as a way to classify and act on a subset of a signaling interface's traffic, also have **rfc2833-mode** and **rfc2833-payload** parameters. The configurations of these parameters overrides the configuration of the same-named parameters on the signaling interface where the session agent resides. You can set the **rfc2833-mode** parameter to **none** for a session agent to revert to the parent signaling interface's two RFC 2833 settings.

ACLI Configuration and Instructions

This section explains how to configure the RFC 2833 mode on a signaling interface and on a session agent configured for that signaling interface. The session agent's configuration takes precedence over the signaling interface, unless the session agent's **rfc2833-mode** is set to **none**. In that case, the signaling interface's configuration is used for applicable traffic.

SIP Interface

To configure the RFC 2833 mode on a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
```
4. If you are adding support for this feature to a pre-existing SIP interface, then you must select the specific configuration instance, using the ACLI **select** command.
5. **rfc2833-mode**—Set this parameter to either **transparent**, **preferred**, or **dual** based upon the behavior you want for this SIP interface.

- transparent—does not add RFC 2833 telephone-event into SDP if not present, and does not prefer.
 - preferred—adds RFC 2833 telephone-event media type into SDP and prefers to use this method for DTMF indication.
 - dual—adds RFC 2833 telephone-event media type into SDP and sends both SDP and signaling-based DTMF indications if possible.
6. **rfc2833-payload**—Set this parameter to the media-type value you wish to use when inserting RFC 2833 telephone-events into an SDP offer. 101 is the generally accepted media type for RFC 2833 telephone-events.
 7. Save and activate your configuration.

H.323 Stack

To configure the RFC 2833 mode on an H.323 stack:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **h323-config** and press <Enter>.


```
ACMEPACKET(session-router)# h323-config
```
4. Type **h323-stack** and press <Enter>.


```
ACMEPACKET(h323-config)# h323-stack
ACMEPACKET(h323-stack)#
```
5. If you are adding support for this feature to a pre-existing H.323 stack, then you must select the specific configuration instance, using the ACLI **select** command.
6. **rfc2833-mode**—Set this parameter to either **transparent**, **preferred**, or **dual** based upon the behavior you want for this SIP interface.
 - transparent—does not add RFC 2833 telephone-event into SDP if not present, and does not prefer.
 - preferred—adds RFC 2833 telephone-event media type into SDP and prefers to use this method for DTMF indication.
 - dual—adds RFC 2833 telephone-event media type into SDP and sends both SDP and signaling-based DTMF indications if possible.
7. Save and activate your configuration.

Session Agent

Session agent RFC 2833 mode configurations override those on the signaling interface where they exit. The **none** parameter is used to defer to the signaling interface.

To configure the RFC 2833 mode on a session agent:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```

3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. Select the session agent where you want this feature.

```
ACMEPACKET(session-agent)# select 1
```
5. **rfc2833-mode**—Set this parameter to either **none**, **transparent**, **preferred**, or **dual** based upon the behavior you want for this SIP interface.
 - **none**—defaults to the behavior of the SIP interface for traffic that matches this session agent.
 - **transparent**—does not add RFC 2833 telephone-event into SDP if not present, and does not prefer.
 - **preferred**—adds RFC 2833 telephone-event media type into SDP and prefers to use this method for DTMF indication.
 - **dual**—adds RFC 2833 telephone-event media type into SDP and sends both SDP and signaling-based DTMF indications if possible.
6. **rfc2833-payload**—Set this parameter to the media-type value you wish to use when inserting RFC 2833 telephone-events into an SDP offer. 101 is the generally accepted media type for RFC 2833 telephone-events.
7. Save and activate your configuration.

Transcoding Policy

To configure a transcoding policy to support DTMF audio tones, as transcoded:

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

```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.

```
configure# media-manager
```
3. Type **transcoding-policy** and press <Enter>.

```
ACMEPACKET(media-manager)# transcoding-policy
ACMEPACKET(transcoding-policy)#
```
4. If you are adding support for this feature to a pre-existing configuration, then you must select the specific configuration instance, using the ACLI **select** command.

```
ACMEPACKET(transcoding-policy)# select
<name>:
1: private
2: public
selection:1
ACMEPACKET(transcoding-policy)#
```
5. **dtmf-in-audio**—Set this parameter to **disabled**, **preferred**, or **dual** based upon how the Net-Net 9200 should support the conversion of signaling messages or RFC 2833 to DTMF Audio tones in the realm where this transcoding policy is active.
 - **disabled**—does not support DTMF audio tones as transcoded in this realm.
 - **preferred**—supports DTMF audio tones as transcoded in this realm.
 - **dual**—supports both transcoded DTMF audio tones and signaling-based DTMF indications if possible.

6. Save and activate your configuration.

Translate Non2833 Event Behavior

To configure the exceptional behavior described in “Identical Inband with Signaling DTMF Transfer Exception” on page 559:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-manager** and press <Enter> to begin configuring this feature.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **translate-non-rfc2833-event**—Set this parameter to enabled to use non-default behavior described in [Override Preferred RFC 2833 \(560\)](#).
5. **translate-non-inband-event**—Set this parameter to enabled to use non-default behavior described in [Override Preferred DTMF Audio \(560\)](#).
6. Save and activate your configuration.

P-dual-info Header Appearance

Customizing the P-Dual-Info header is performed globally from the sip config.

To configure how the P-dual-info header appears:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. Type **options** followed by a <Space>.
5. After the <Space>, type the P-Dual-Info header information in the following format:


```
+dual-info="<header-name>"
```

 For example:


```
ACMEPACKET(sip-config)# options dual-info=P-Dual-Info
```
6. Save your work using the ACLI **done** command.

RFC 2833 Customization

RTP Timestamp

As a media flow with injected RFC 2833 telephone-event packets exits the Net-Net SBC, the newly generated RTP packets are timestamped in one of two ways. If RFC 2833 generation is preformed in the NPs, as explained in [DTMF Transfer Hardware](#)

[Processing \(562\)](#), the default method of creating the timestamp for a generated RFC 2833 packet is to use the previous RTP packet's timestamp and add 1.

Alternatively the Net-Net 9200 can estimate the actual time that the injected RFC 2833 telephone-event packet will leave the system and use that. This method tags the injected DTMF indication packet more accurately than the *previous packet + 1* method. As an additional bonus, the packet's checksum is regenerated. This alternate timestamp creation behavior is configured by setting the **rfc2833-timestamp** parameter to enabled.

When RFC 2833 telephone-event generation is preformed by the TCUs, the packets' timestamps are the set to the time that the packets leave the Net-Net 9200.

RFC 2833 telephone-event duration intervals

If an incoming SIP INFO message's DTMF indication duration is unspecified, the Net-Net SBC uses a default 250 ms duration for the generated RFC 2833 telephone-event. Otherwise, the SIP INFO's specified event duration is used. RFC 2833 telephone-event packets are still generated at 50 ms intervals upon egress. At the conclusion of the DTMF indication, the three end-event packets are sent. The packet arrangement when the user presses the digit "5" for 160ms, with the default 50ms interval follows:



When either no DTMF event duration is specified, or the event duration is less than the 50ms default minimum, you can set the default RFC 2833 telephone-event duration using **default-2833-duration** parameter in the media manager configuration. This is the value that the Net-Net SBC uses for the duration of a telephone event when none is specified in the incoming message. The **default-2833-duration**'s valid range is 50-5000ms. The Net-Net SBC also uses this configured value when it receives an H.245UII or SIP INFO message with a duration less than the minimum signal duration.

You can configure the minimum duration at which RFC 2833 telephone-events are generated by the Net-Net SBC using the **min-signal-duration** option in the media manager configuration, thus changing the lower threshold of the **default-2833-duration** parameter from 50 ms to your own value. If the duration the Net-Net SBC receives is less than the threshold, it uses the value configured in the **default-2833-duration** parameter.

Note: Timestamp changes and duration changes only take effect when the 2833 timestamp (**rfc-2833-timestamp**) is enabled in the media manager configuration. If you enable the **rfc-2833-timestamp** parameter, but do not configure the **default-2833-duration** parameter, the **default-2833-duration** parameter defaults to 100 ms.

RFC 2833 End Packets

When the Net-Net SBC generates RFC 2833 telephone-event packets, they are forwarded from the egress interface every 50 ms by default. Each packet includes the digit and the running total of time the digit is held. Thus DTMF digits and events are sent incrementally to avoid having the receiver wait for the completion of the event.

At the conclusion of the signaled event, three end packets stating the total event time are sent. This redundancy compensates for RTP being an unreliable transport protocol.

You can configure your Net-Net SBC to generate either the entire start-interim-end RFC 2833 packet sequence or only the last three end 2833 packets for non-signaled digit events using the **rfc2833-end-pkts-only-for-non-sig** parameter. Based on the diagram in the previous section, if the parameter were enabled, the RFC 2833 telephone-event packets for the same event would appear as represented by the following graphic.



ACLI Instructions and Examples

To configure RFC 2833 customization:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-manager** and press <Enter> to begin configuring this feature.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **rfc2833-timestamp**—Set this parameter to enabled to use the estimated real departure timestamp on an injected RFC 2833 telephone-event packet.
5. **default-2833-duration**—Set this parameter to the default value you wish to use when the Net-Net SBC creates or generates DTMF-indication messages.
6. **rfc2833-end-pkts-only-for-non-sig**—Set this parameter to enabled for the Net-Net SBC to only send the three RFC 2833 telephone-event end-packets to indicate a total event duration, rather than the running total from time=0.
7. **options**—Set the options parameter by typing **options**, a <Space>, the option name **min-signal-duration=x** (where x is the value in milliseconds you want to use for the threshold) with a “plus” sign in front of it. Then press <Enter>.
8. Save and activate your configuration.

Introduction

This chapter describes the Net-Net 9200's NAT ALG services. It describes both and provides information for how to configure them.

NAT ALG

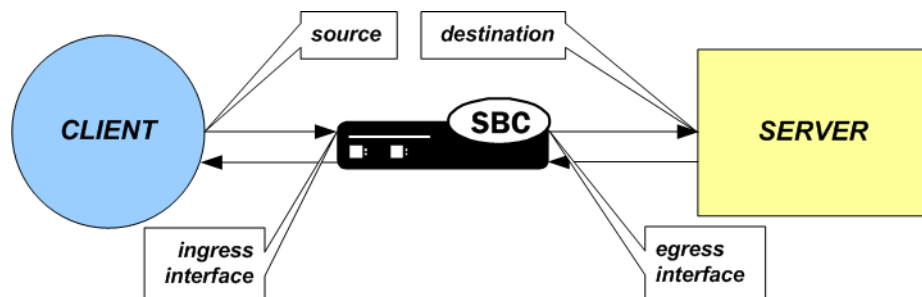
The Net-Net 9200 now supports Network Address and Port Translation (NAPT) and Trivial File Transfer Protocol (TFTP) functionality over media interfaces, collectively known as Network Address Translation (NAT) ALG. The NAT ALG feature is implemented as an extension of the static flow feature and is configured with seven new parameters.

In some applications, the Net-Net 9200 acts as an intermediary device, positioned between endpoints located in an access network and application servers located in a backbone network. The Net-Net 9200's NAT ALG feature enables these endpoints to use non-VoIP protocols, such as TFTP and HTTP, to access servers in a provider's backbone network to obtain configuration information.

NAT ALG parameters support RTC and can be dynamically reconfigured. The active NAT ALG configuration can be replicated on the standby 9200 in an HA configuration.

NAPT

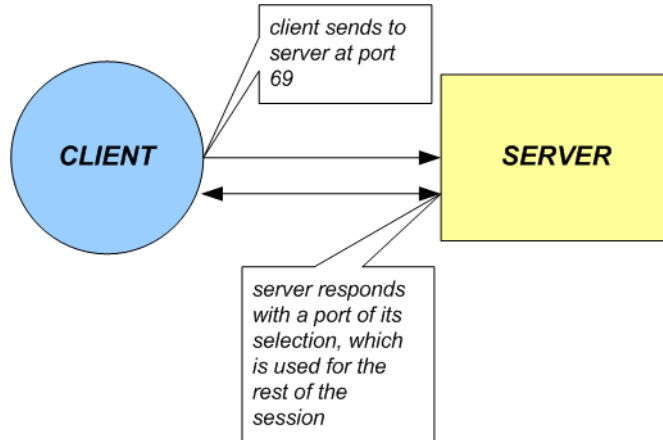
The NAPT ALG functionality is the same as that found in commercially available enterprise and residential NAT devices. The Net-Net 9200 watches for packets entering a media interface that match source and destination IP address criteria. Matching packets are then redirected out of the egress interface, through a specified port range, toward a destination address.



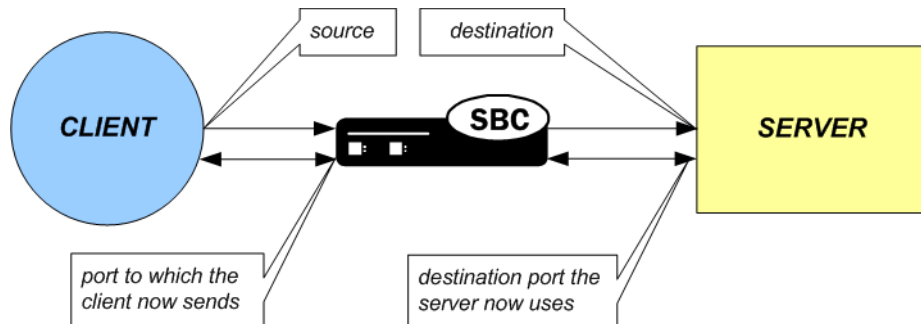
TFTP

The TFTP ALG is implemented as an extension of the NAT ALG. It works slightly differently than traditional NAPT. In a TFTP session, the first packet is sent from a source endpoint to port 69 on the TFTP server. The TFTP server responds from

another port. This port, from which the TFTP response originates, is used for the remainder of the TFTP session.



To act as a TFTP ALG, the Net-Net 9200 will latch on the first return packet from the server to learn the server's port. The ingress-side destination port of the Net-Net 9200 is changed to reflect the new communications port for the TFTP session. This process takes place without any user intervention.



Summary of NAT ALG Settings

The NAT ALG is set by configuring all of the parameters contained within the **static-flow** element.

The ingress IP address criteria is set first. These parameters are applicable to traffic entering the ingress side of the Net-Net 9200.

- **in-realm-id**—The access realm, where endpoints are located.
- **in-source**—The source network in the access realm where the endpoints exist. This parameter is entered as an IP address and netmask in slash notation to indicate a range of possible IP addresses.
- **in-destination**—The IP address and port pair where the endpoints send their traffic. This is usually the IP address and port on a Net-Net 9200 network interface that faces the access realm.

The egress IP address criteria is entered next. These parameters determine how traffic is re-sourced as it leaves the Net-Net 9200 and enters the backbone network.

- **out-realm-id**—The backbone realm, where servers are located.
- **out-source**—The IP address on the network interface of the Net-Net 9200 where traffic exits the 9200 into the backbone realm. Do not enter a port for this parameter.
- **out-destination**—The IP address and port pair destination of the traffic this NAT ALG defines. This is usually a server in the backbone realm.

The type of NAT ALG and traffic it affects is set next.

- **protocol**—The protocol this NAT ALG affects. Valid entries for this parameter are TCP, UDP, ICMP, ALL.
- **alg-type**—The type of NAT ALG. Set this to NATP.

The port range for port re-sourcing as traffic affected by the NAT ALG exits the egress side of the Net-Net 9200 is set next.

- **start-port**—The starting port the NAT ALG uses as it re-sources traffic on the egress side of the Net-Net 9200.
- **end-port**—The ending port the NAT ALG uses as it re-sources traffic on the egress side of the Net-Net 9200.

The flow timers and bandwidth policing parameters are set next.

- **flow-time-limit**—Total session time limit in seconds. The default is 0; no limit. This field is not applicable if **alg-type** is none.
- **initial-guard-timer**—Initial flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit. This field does not apply if **alg-type** is set to none.
- **susbsq-guard-timer**—Subsequent flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit. This field does not apply if **alg-type** is set to none.
- **average-rate-limit**—Sustained rate limit in bytes per second for the static flow and any dynamic ALG flows. The default is 0; no limit.

TFTP ALG

The TFTP ALG is configured identically to the NAT ALG, except for the following parameter.

alg-type—The type of NAT ALG. Set this to TFTP.

Configuring the NAT ALG

Configure the NAT ALG in the static flow configuration element.

To configure the NAT ALG:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the **media-manager** path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **static-flow** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# static-flow
```

From this point, you can configure media policing parameters.

4. **in-realm-id**—Enter the defined source realm where traffic for this NAT ALG originates.
5. **in-source**—Enter the network that defines where the incoming traffic originates. This parameter is entered as an IPv4 address and netmask in slash notation.
6. **in-destination**—Enter the ingress side's destination. This is the IPv4 address and port of the network interface in the ingress realm.
7. **out-realm-id**—Enter the defined realm where traffic leaving this NAT ALG exits the Net-Net 9200.
8. **out-source**—Enter the egress IPv4 address. This is the IPv4 address of the network interface where traffic subject to the NAT ALG you are defining leaves the Net-Net 9200. Do not enter a port number for this parameter.
9. **out-destination**—Enter the IPv4 address and port number of the server or other destination this NAT ALG redirects traffic toward.
10. **protocol**—Enter the protocol this NAT ALG acts upon. Valid values are: TCP, UDP, ICMP, ALL.
11. **alg-type**—Enter NAT to create a NAT ALG.
12. **start-port**—Enter the beginning port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this NAT ALG redirects.
13. **end-port**—Enter the ending port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this NAT ALG redirects.
14. **flow-time-limit**—Enter the total time limit for an NAT flow in seconds. The default is 0; no limit.
15. **initial-guard-timer**—Enter the initial guard timer value in seconds. The default is 0; no limit.
16. **subsq-guard-timer**—Enter the subsequent guard timer value in seconds. The default is 0; no limit.
17. **average-rate-limit**—Enter a maximum sustained rate limit in bytes per second for this NAT ALG. The default is 0; no limit.

The following example shows a **static-flow** configuration element configured for a NAT ALG.

in-realm-id	access
in-source	172.16.0.0/16
in-destination	172.16.1.16:23
out-realm-id	backbone
out-source	192.168.24.16
out-destination	192.168.24.95:23
protocol	TCP
alg-type	NAPT
start-port	11000
end-port	11999
flow-time-limit	0
initial-guard-timer	60
subsq-guard-timer	60
average-rate-limit	0
peak-rate-limit	0
maximum-burst-size	0

Configuring the TFTP ALG

You configure the TFTP ALG in the static flow configuration element.

To configure the TFTP ALG:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the **media-manager** path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **static-flow** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# static-flow
```

From this point, you can configure media policing parameters.
4. **in-realm-id**—Enter the defined source realm where traffic for this TFTP ALG originates.
5. **in-source**—Enter the network that defines where the incoming traffic originates. This parameter is entered as an IPv4 address and netmask in slash notation.
6. **in-destination**—Enter the ingress side's destination. This is the IPv4 address and port of the network interface in the ingress realm.
7. **out-realm-id**—Enter the defined realm where traffic leaving this NAT ALG exits the Net-Net 9200.
8. **out-source**—Enter the egress IPv4 address. This is the IPv4 address of the network interface where traffic subject to the TFTP ALG you are defining leaves the Net-Net 9200. Do not enter a port number for this parameter.
9. **out-destination**—Enter the IPv4 address and port number of the TFTP server this TFTP ALG redirects traffic toward. The port number for this address will be 69.
10. **protocol**—Enter the protocol this TFTP ALG acts upon. Valid values are: TCP, UDP, ICMP, ALL.
11. **alg-type**—Enter TFTP to create a TFTP ALG.
12. **start-port**—Enter the beginning port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this TFTP ALG redirects.
13. **end-port**—Enter the ending port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this TFTP ALG redirects.
14. **flow-time-limit**—Enter the total time limit for an TFTP flow in seconds. The default is 0; no limit.
15. **initial-guard-timer**—Enter the initial guard timer value in seconds. The default is 0; no limit.
16. **subsq-guard-timer**—Enter the subsequent guard timer value in seconds. The default is 0; no limit.
17. **average-rate-limit**—Enter a maximum sustained rate limit in bytes per second for this TFTP ALG. The default is 0; no limit.

The following example shows a **static-flow** configuration element configured for a TFTP ALG.

```
in-realm-id          access
      in-source      172.16.0.0/16
      in-destination  172.16.1.16:23
```

out-realm-id	backbone
out-source	192.168.24.16
out-destination	192.168.24.95:23
protocol	TCP
alg-type	TFTP
start-port	11000
end-port	11999
flow-time-limit	0
initial-guard-timer	60
subsq-guard-timer	60
average-rate-limit	0
peak-rate-limit	0
maximum-burst-size	0

Introduction

The Net-Net SBC's DNS Application Layer Gateway (ALG) feature enables the Net-Net SBC to support the appearance of multiple DNS servers in one (or more) realms from the perspective of a DNS client in the access realm.

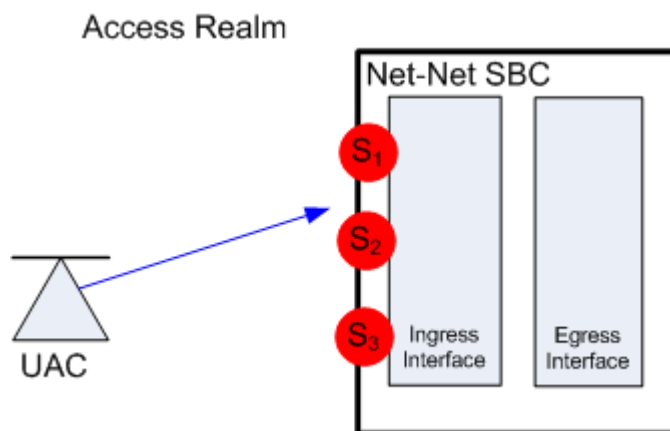
Overview

When a DNS client and DNS server exist in separate realms, the Net-Net SBC acts as an intermediary by enabling DNS transactions to occur across these realms. The Net-Net 9000's DNS ALG allows a DNS client in one realm, such as a UAC, to send DNS requests to the Net-Net SBC. After applying the proper NAT, these requests are then forwarded through the Net-Net SBC to DNS servers in a different realm.

Once the configured DNS server is chosen, the Net-Net SBC acts as a DNS client on behalf of the real client (the UAC) in the access network. The SBC sends the request to a real DNS server, and finally returns the response to the originating UAC.

Client (Access) Realm

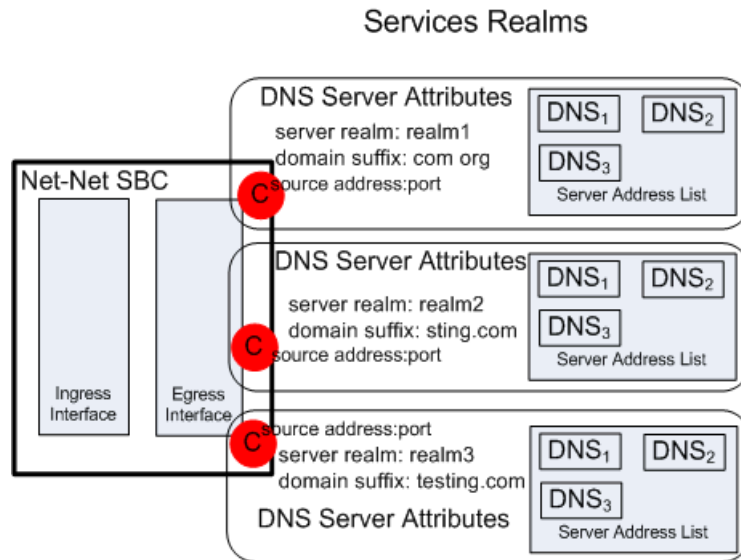
There can only be one ALG instance per realm. In an access realm, the DNS ALG is configured with multiple DNS ALG server IP addresses, that can only listen on UDP port 53. These IP addresses that act as DNS servers are configured in the client-address-list parameter. These server address and ports are depicted in the below diagram as S1, S2, S3.



DNS Server Realms

After configuring the access-side realms, you need to configure the service realms, where the DNS servers reside. Each realm that you configure corresponds to a DNS ALG, which will serve a specific set of domain suffixes (com, org, testing.com). From the Net-Net SBC's point of view, it creates a DNS client (on behalf of the real DNS client) that exists in a service realm. The DNS client is configured as a single IP address and port in the source-address and source-port parameters in the server DNS attributes configuration element. The DNS servers in this service realm are configured in the server-address-list parameter, which can take multiple values.

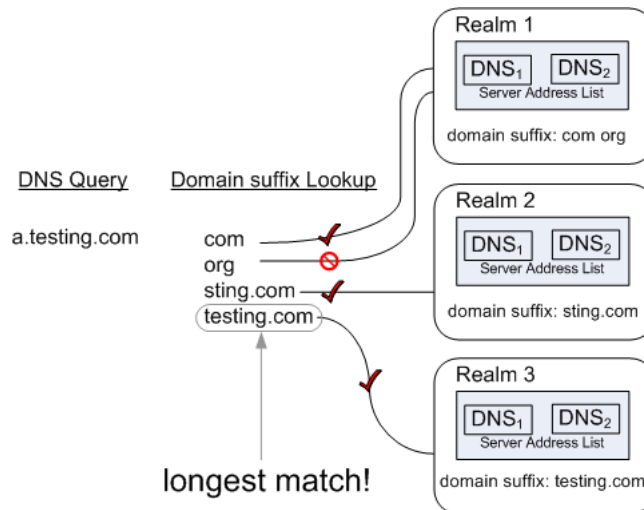
Finally, you configure one or more domain suffix parameters that indicate the DNS domains that these DNS servers can resolve.



DNS Realm Target Selection

The Net-Net SBC decides which realm, and ultimately the specific DNS server, to send a DNS request to by examining the domain name in the request. This domain name is compared to the domain suffixes of all configured server-dns-attribute entries. The longest match made between the UAC's query and the domain suffix in any of the server-dns-attribute entries' is used to select the realm and DNS server group (server-address-list) to send the DNS request to.

In the following image, the domain suffix testing.com was the longest match for the example DNS query (a.testing.com). Because testing.com is configured in Realm3's server-dns-attributes configuration element, the DNS servers listed in that configuration will be used to resolve the query.



DNS Server Selection

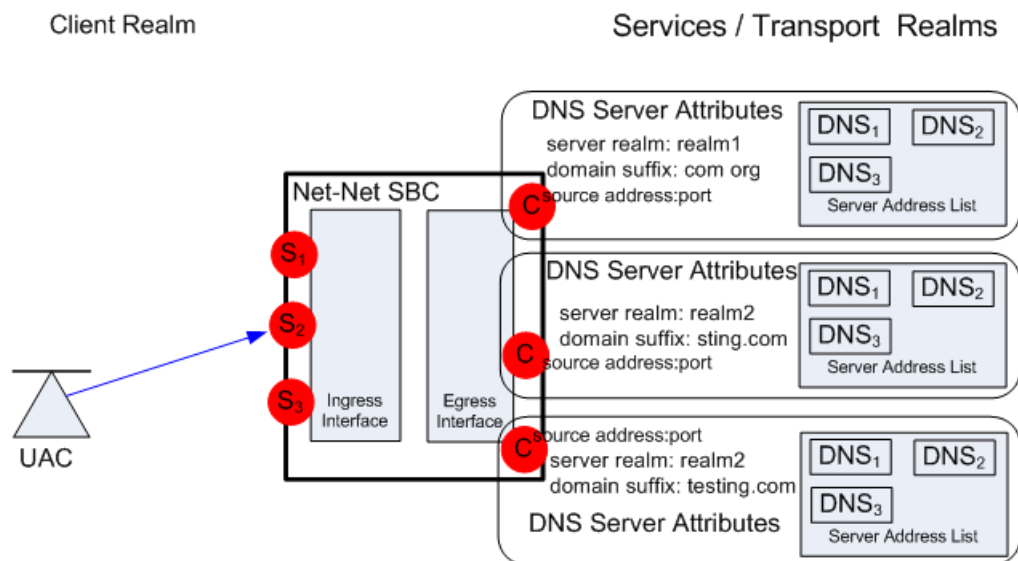
Now that the DNS ALG has chosen a realm/DNS-server-attributes configuration, the egress realm and source IP address:port are also known. The Net-Net SBC still

must choose a specific DNS server from the list of available servers (server address list) in that realm.

The Net-Net SBC selects a specific DNS server based on its index. When configuring the access-side client address list, you configure multiple DNS server IP addresses. These are indexed from the first to last configured. Similarly, in the service realm, the list of DNS server addresses is indexed from the first to the last configured.

When the Net-Net SBC receives a DNS query on the access side, the index of the IP address on which the query was received is noted. After the Net-Net SBC makes a DNS domain match and chooses a realm for forwarding the DNS request into, the actual DNS server in that realm must be chosen. Based on the access-side receiving IP address's index, the Net-Net SBC chooses the corresponding services-side DNS server, by index.

For example, in the following diagram, if the Net-Net SBC receives the DNS query on S2, and determines the query should be sent to Realm3, it will forward the query to DNS2.



If there is no corresponding DNS server, or the corresponding DNS server does not respond to the query, the Net-Net SBC forwards a DNS server not found error message back to the originating UAC.

Note: Typically only one client realm server address is used.

Address Translation

The Net-Net SBC can perform address translation on the replies (DNS responses) it forwards back to the originating UAC. In order to configure this, you must set a server address prefix and a client address prefix. You configure these parameters in any DNS server attributes configuration element.

The network portion of the server address prefix, configured as an IP address and subnet mask in slash notation, is matched against the network portion of a DNS server's response. If there is a positive match, then the client address prefix translation is applied to the DNS response on the egress side.

The client address prefix, configured as an IP address and subnet mask in slash notation, replaces the network portions of the IP addresses returned in the DNS response's answer section as they are forwarded back to the UAC.

Configuring DNS ALG Service

You can access the configuration parameters for DNS ALG service using the ACLI.

Configuring DNS ALG service requires that you carry out two main procedures:

- Setting the name, realm, and DNS service IPv4 interfaces
- Setting the appropriate parameters for DNS servers to use in other realms

Before You Configure

Before you begin to configure DNS ALG service on the Net-Net SBC, complete the following steps.

1. Configure the client realm in the DNS configuration profile.
2. Configure the server realms that contains the DNS servers.
3. Determine the domain suffixes for the network where the DNS servers are located so that you can enter them in the domain suffix parameter.
4. Devise the NAT scheme that you want to use when the DNS reply transmits the Net-Net SBC.

ACLI Instructions and Examples

This section explains how to configure the name of the DNS ALG service you are configuring and set its realm.

To add DNS ALG service:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **dns-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# dns-config
ACMEPACKET(dns-config)#
```

From this point, you can configure DNS ALG parameters and access this configuration's DNS server subelement. To view all DNS ALG service parameters and the DNS server subelement, enter a ? at the system prompt.

```
dns-config
  client-realm
  description dns-alg1
  client-address-list
  last-modified-date 2005-02-15 10:50:07
  server-dns-attributes
    server-realm
    domain-suffix
    server-address-list
    source-address
    source-port 53
    transaction-timeout 10
```

```

address-translation
  server-prefix          10.3.0.0/16
  client-prefix          192.168.0.0/16

```

Identity, Realm, and Interface Addresses

To configure the identity, realm, and IPv4 interface addresses for your DNS ALG profile:

1. **description**—Set a name for the DNS ALG profile using any combination of characters entered without spaces. You can also enter any combination with spaces if you enclose the whole value in quotation marks. For example: “DNS ALG service.”
2. **client-realm**—Enter the name of the realm from which DNS queries are received. If you do not set this parameter, the DNS ALG service will not work.
3. **client-address-list**—Configure a list of one or more addresses for the DNS server interface. These are the addresses in the access realm to which DNS clients send queries.

To enter one address in this list, type **client-address-list** at the system prompt, a <Space>, the IPv4 address, and then press <Enter>

```
ACMEPACKET (dns-config)# client-address-list 192.168.0.2
```

To enter more than one address in this list, type **client-address-list** at the system prompt, and a <Space>. Then type an open parenthesis ((), each IPv4 address you want to use separated by a <Space>, and closed parenthesis ()), and then press <Enter>.

```
ACMEPACKET (dns-config)# client-address-list (192.168.0.2 196.168.1.1 192.168.1.2)
```

DNS Server Attributes

To configure attributes for the DNS servers that you want to use in the DNS ALG profile:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **dns-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# dns-config
```
4. Type **server-dns-attributes** and then press <Enter>.


```
ACMEPACKET(dns-config)# server-dns-attributes
```

From this point, you can configure DNS server parameters. To see all parameters for the DNS server, enter a ? at the system prompt.
5. **server-realm**—Enter the name of the realm in which the DNS server is located. This value is the name of a configured realm.
6. **domain-suffix**—Enter a list of one or more domain suffixes to indicate the domains you want to serve. These values are matched when a request is sent to a specific DNS server. If you leave this list empty (default), then your configuration will not work.

Note: If you want to use a wildcard value, you can start your entry to an asterisk (*) (e.g. *.com). You can also start this value with a dot (e.g., .com).

To enter one address in this list, type `client-address-list` at the system prompt, a `<Space>`, the domain suffix, and then press `<Enter>`

```
ACMEPACKET (server-dns-attributes)# domain-suffix acmepacket.com
```

To enter more than one address in this list, type `domain-suffix` at the system prompt, and a `<Space>`. Then type an open parenthesis (`(`), each IPv4 address you want to use separated by a `<Space>`, and closed parenthesis (`)`), and then press `<Enter>`.

```
ACMEPACKET (server-dns-attributes)# domain-suffix (acmepacket.com
acmepacket1.com acmepacket2.com)
```

7. **server-address-list**—Enter a list of one or more DNS IPv4 addresses for DNS servers. These DNS servers can be used for the domains you specified in the domain suffix parameter. Each domain can have several DNS servers associated with it, and so you can populate this list with multiple IPv4 addresses. If you leave this list empty (default), your configuration will not work.
8. **source-address**—Enter the IPv4 address for the DNS client interface on the Net-Net SBC. If you leave this parameter empty (default), your configuration will not work.
9. **source-port**—Enter the number of the port for the DNS client interface on the Net-Net SBC. You can leave this parameter set to 53 (default), or enter a new port number.
10. **transaction-timeout**—Enter the time in seconds that the ALG should keep information to map a DNS server response back to the appropriate client request. After the transaction times out, further response to the original request will be discarded.

You can leave this parameter set to its default of 10 seconds.

11. **address-translation**—Enter a list of address translations that define the NAT function for the DNS servers' response.

You can access the NAT parameters for the DNS servers by typing `address-translation` and pressing enter within the server DNS attributes configuration.

```
ACMEPACKET(server-dns-attributes)# address-translation
ACMEPACKET(address-translation)#
```

To configure the NAT, enter two values:

- **server-prefix**: address/prefix that will be returned by the DNS server
- **client-prefix**: address/prefix that to which a response is returned

Each of these is a two-part value:

- IPv4 address
- Number of bits indicating how much of the IPv4 address to match

If you do not specify the number of bits, then all 32 bits of the IPv4 address will be used for matching. If you set the number of bits to 0, then the address will simply be copied.

For example, if you set the server prefix to 10.3.17.2/16 and the client prefix to 192.168.0.0/16, then the Net-Net SBC will return an address of 192.168.17.2 to the DNS client.

```
ACMEPACKET(address-translation)# server-prefix 10.3.17.2/16
ACMEPACKET(address-translation)# client-prefix 192.168.0.0/16
```

DNS Transaction Timeout

To provide resiliency during DNS server failover, you can now enable a transaction timeout for DNS servers. If you have endpoints that are only capable of being configured with a single DNS server, this can allow DNS queries to be sent to the next configured server—even when contacting the Net-Net SBC's DNS ALG on a single IP address. So when the first server in the list times out, the request is sent to the next server in the list.

The Net-Net SBC uses the transaction timeout value set in the **dns-server-attributes** configuration (part of the **dns-config**).

ACLI Instructions and Examples

To enable the DNS transaction timeout:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-manager** and press <Enter>.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **dnsalg-server-failover**—Change this parameter from **disabled** (default) to **enabled** to allow DNS queries to be sent to the next configured server—even when contacting the Net-Net SBC's DNS ALG on a single IP address. So when the first server in the list times out, the request is sent to the next server in the list. The Net-Net SBC uses the transaction timeout value set in the **dns-server-attributes** configuration (part of the **dns-config**).
5. Save and activate your configuration.

DNS Server Failover

When a **dns-config**'s configured server is unreachable via the network, i.e., no response is received on a query, the Net-Net SBC can failover to another server in the defined **server-address-list**.

The Net-Net SD attempts to reach a DNS server upon a client request. After the first failed attempt to reach a DNS server, it is placed in a Standby state, in which it stays for 30 seconds. Within this 30 seconds, the Net-Net SD does not contact the server for subsequent queries. After the 30 seconds pass, the Net-Net SD sets the DNS server back to an Active state and directs queries to it. If this sequence occurs 5 consecutive times, the Net-Net SD sets the DNS server to Out of Service (OOS). All subsequent queries that would be sent to the OOS servers are instead sent to the next Active server in the **server-address-list**. The Net-Net SD sets OOS DNS servers back to Active after 600 seconds and restarts this process.

DNS statistics

DNS resolution statistics are available from the **show dns server** command.

11 Session Routing and Load Balancing

Introduction

This chapter explains how to configure session routing and load balancing for SIP services. It contains information about configuring session agents and session agent groups, as well as local policies that can be used for routing SIP signals.

Routing Overview

This section provides an overview of routing SIP sessions when using the Net-Net 9200. The Net-Net 9200 chooses the next hop through the network for each SIP session based on information received from routing policies and constraints. Routing policies can be as simple as routing all traffic to a proxy or routing all traffic from one network to another. Routing policies can also be more detailed, using constraints to manage the volume and rate of traffic that can be routed to a specific network. For example, you can manage volume and rate of traffic to enable the Net-Net 9200 to load balance and route around softswitch failures.

When a call request arrives at the Net-Net 9200, a decision making process then occurs to determine whether the message is coming from a session agent. If so, the Net-Net 9200 checks whether that session agent is authorized to make the call. Local policy is then checked to determine where to send the message on to.

Session Agents, Session Groups, and Local Policy

You can use session agents, session agent groups and local policies to define routing. However, using session agents and session agent groups is not required.

- Session agent—Defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes.
- Session agent group (SAG)—Can contain individual session agents and other session agent groups. Members of a SAG are logically equivalent (although they might vary in their individual constraints) and can be used interchangeably.

You apply an allocation strategy to the SAG to allocate traffic across the group members. Session agent groups also assist in load balancing among session agents.

- Local policy—Indicates where session request messages, such as SIP INVITEs, are routed and/or forwarded. You use a local policy to set a preference for selecting one route over another.

Realms and Routing

Realms are also used in routing decisions when a Net-Net 9200 communicates with multiple network elements over a shared intermediate connection. Defining realms allows sessions to go through a connection point between the two networks. See the *Realms and Nested Realms* chapter for additional details.

When you configure a realm, you give it an identifier, which stores the name of the realm associated with the Net-Net system. The realm identifier value is also needed when you configure session agents and local policies. You can associate a realm with a session agent to identify the realm for sessions coming from or going to the session agent. You also need the realm identifier when you configure local policy to identify the egress realm (realm of the next hop).

About Session Agents

A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. Service elements such as gateways, softswitches, and gatekeepers are defined automatically within the Net-Net 9200 as session agents. For each session agent, concurrent session capacity and rate attributes can be defined. You can group session agents together into session agent groups and apply allocation strategies to achieve traffic load balancing.

You can configure a set of attributes and constraints for each session agent to support the following:

- Session access control—Net-Net 9200 only accepts requests from configured session agents
- Session admission control (concurrent sessions)—Net-Net 9200 limits the number of concurrent inbound and outbound sessions for any known service element
- Session agent load balancing—Session agents are loaded based on their capacity and the allocation strategy specified in the session agent group
- Session (call) gapping—Net-Net 9200 polices the rate of session attempts to send to and receive from a specific session agent

SIP Session Agents

SIP session agents can include the following:

- Softswitches
- SIP proxies
- Application servers
- SIP gateways
- SIP endpoints

In addition to functioning as a single logical next hop for a signaling message (for example, where a SIP INVITE is forwarded), session agents can provide information about next or previous hops for packets in a SIP agent, including providing a list of equivalent next hops.

You can use the session agent to describe one or more SIP next or previous hops. Through the configured carriers list, you can identify the preferred carriers to use for traffic coming from the session agent. This set of carriers will be matched against the local policy for requests coming from the session agent. You can also set constraints for specific hops.

SIP Session Agent Status Based on SIP Response

The Net-Net SBC can now take session agents out of service based on SIP response codes that you configure, and you can also configure SIP response codes that will keep the session agent in service.

In prior releases, the Net-Net SBC determines session agents' health by sending them ping messages using a SIP method that you configure. Commonly, the method is an OPTIONS request. If it receives any response from the session agent, then the Net-Net SBC deems that session agent available for use.

However, issues can arise when session agents are administratively out of service, but able to respond to OPTIONS requests. A session agent like this might only respond with a 200 OK when in service, and send a 4xx or 5xx message otherwise.

The new session agent status feature lets you set the SIP response message that either takes a session agent out of service or allows it to remain in service when it responds to the Net-Net SBC's ping request.

Details of this feature are as follows:

- The Net-Net SBC only considers a session agent in service when it responds to a request method you set with the final response code that you also set. If a final response code is set, then provisional responses are not used for determining whether or not to take a session agent out of service. If the Net-Net SBC receives a final response code that does not match the session agent configuration, it treats the session agent as though it had not responded.
- The Net-Net SBC takes a session agent out of service when it receives an error response for dialog creating request with a response code listed in the new **out-service-response-codes** parameter.
- In the case where a the session agent's response has a Retry-After header, the Net-Net SBC tries to bring the session agent back into service after the period of time specified in the header. To do so, it sends another ping request.

How It Works

There are two lists you can configure in the session agent configuration to determine status:

- In-service list—Set in the CLI **ping-in-service-response-codes** parameter, this list defines the response codes that keep a session agent in service when they appear in its response to the Net-Net SBC's ping request. Furthermore, the Net-Net SBC takes the session agent out of service should a response code be used that does not appear on this list.
- Out-of-service list—Set in the CLI **out-service-response-codes** parameter, this list defines the response codes that take a session agent out of service when they appear in its response to the Net-Net SBC's ping request.

When the Net-Net SBC receives a session agent's response to its ping request, it first checks to see if there is an in-service list of responses configured for that session agent. If the list is configured and the Net-Net SBC determines that there is a match, the session agent is deemed in service. Otherwise it takes the session agent out of service. In this way, the in-service list takes precedence over the out-of-service list. If you configure the in-service list, then the Net-Net SBC ignores the out-of-service list.

If there is no list of in-service responses for the session agent, then the Net-Net SBC checks the out of service list. If it is configured and the Net-Net SBC determines that

there is a match, the Net-Net SBC removes that session agent from service. If there is no match, then the session agent is deemed in service.

ACLI Instructions and Examples

Remember when you are configuring this feature that the Net-Net SBC first checks for the presence of the in-service list. If it finds one, then it ignores the out-of-service list. Therefore, to use the out-of-service list, do not configure the in-service list.

For both parameters, values (response codes) are entered in comma-separated lists, and as ranges, for example:

```
ACMEPACKET(session-agent)# ping-in-service-response-codes
404,407,484-486
```

To configure session agent status based on SIP response code(s):

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.
4. **ping-in-service-response-codes**—Enter the list of response codes that keep a session agent in service when they appear in its response to the Net-Net SBC's ping request. The Net-Net SBC takes the session agent out of service should response code be used that does not appear on this list.
5. **out-service-response-codes**—Enter the list defines the response codes that take a session agent out of service when they appear in its response to the Net-Net SBC's ping request. The Net-Net SBC ignores this list if an in-service list exists.
6. Save and activate your configuration.

Managing Session Agent Traffic

The Net-Net 9200 monitors availability, session load, and session rate for each session agent in real time. The session agent's state is determined by its performance relative to the constraints applied to it and its availability.

The following table lists the conditions that cause the Net-Net 9200 to suspend the routing of traffic to a session agent, along with the criteria for restoring the route.

Constraint Condition	SIP Criteria	Action	Criteria for Resuming
Maximum sessions exceeded	Maximum concurrent SIP sessions exceeded.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum sessions value.
Maximum outbound sessions exceeded	Maximum concurrent outbound SIP sessions exceeded.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum outbound sessions value.
Maximum burst rate exceeded	Maximum burst rate exceeded in current window.	Session agent is declared in constraint violation state.	Burst rate in subsequent window drops below maximum burst rate.
Maximum sustained rate exceeded	Maximum sustained rate exceeded in current window.	Session agent is declared in constraint violation state.	Sustained rate in subsequent window drops below the maximum sustained rate.
Session agent unavailable or unresponsive	SIP transaction expire timer expires for any out-of-dialogue request. For example, INVITE, REGISTER, or ping.	Session agent is declared in constraint violation state or out-of-service. The time to resume timer starts.	Time to resume timer expires and the Net-Net 9200 declares the session agent in-service. or Session agent responds to subsequent pings (SIP only).

About Session Agent Groups

Session agent groups can contain individual session agents and other session agent groups. Members of a session agent group are logically equivalent (although they might vary in their individual constraints) and can be used interchangeably. You can apply allocation strategies to session agent groups.

Examples of session agent groups include the following:

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

Session agent group members do not need to reside in the same domain, network, or realm. The Net-Net 9200 can allocate traffic among member session agents regardless of their location. It uses the allocation strategies configured for a SAG to allocate traffic across the group members.

Allocation strategies include the following:

Allocation Strategy	Description
Hunt	Net-Net 9200 selects the session agents in the order in which they are configured in the SAG. If the first agent is available, and has not exceeded any defined constraints, all traffic is sent to the first agent. If the first agent is unavailable, or is in violation of constraints, all traffic is sent to the second agent. And so on for all session agents in the SAG. When the first agent returns to service, the traffic is routed back to it.
Round robin	Net-Net 9200 selects each session agent in the order in which it is configured, routing a session to each session agent in turn.
Least busy	Net-Net 9200 selects the session agent with the least number of active sessions, relative to the maximum outbound sessions or maximum sessions constraints (lowest percent busy) of the session agent.
Proportional distribution	Session agents are loaded proportionately based upon the respective maximum session constraint value configured for each session agent.
Lowest sustained rate	Net-Net 9200 routes traffic to the session agent with the lowest sustained session rate, based on observed sustained session rate.

You apply allocation strategies to select how session agents that belong to the group are used. For example, if you apply the Hunt strategy session agents are selected in the order in which they are listed.

Throttled Call Teardown to/from Session Agent

When a session agent goes OOS, the calls associated with that session agent are not immediately torn down. The calls are ultimately torn down either by the remote endpoint explicitly ending the call or by flowguard timers expiring.

You can configure the Net-Net SBC to explicitly tear down all calls associated with a session agent when it goes OOS. To mitigate any exceptionally high CPU load placed on the Net-Net SBC when it tears down all calls associated with the downed SA, internal processes will throttle the tear-down rate commensurate with the CPU load.

Calls that are already in progress before this option has been configured and activated are not subject to throttled teardown. For example, if this option is activated after 10 calls are in progress, then another 15 calls are added, all going through a single session agent. If the session agent goes OOS, only the second 15 calls are actively torn down.

Accounting Considerations

The Net-Net SBC writes the stop time that the calls are actually torn down and not the time when the SA went OOS for the disconnect time in the CDR. Media is stopped at the same time too.

To indicate cases like this, where the call was terminated because the associated session agent went OOS, the Net-Net SBC includes the following combination of release/termination causes in the CDR:

Acct-Terminate-Cause = NAS-ERROR

Acme-Disconnect-Cause = 8

ACLI Instructions

To configure throttled tear down of calls to/from an OOS SA:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **preempt-calls-on-oos**, and then press <Enter>.


```
ACMEPACKET(session-agent)# options +preempt-calls-on-oos
```

If you type **options preempt-calls-on-oos**, you will overwrite any previously configured options. In order to append the new option to the **session-agent**'s options list, you must prepend the new option with a "plus" sign as shown in the previous example.
5. Save and activate your configuration.

Local Policy

Local policy lets you indicate where session requests, such as SIP INVITES, should be routed and/or forwarded. You use a local policy to set a preference for selecting one route over another. The local policy contains the following information that affects the routing of the SIP signaling messages:

- Information in the From header

Information in the message's From header is matched against the entries in the local policy's from address parameter to determine if the local policy applies.
- List of configured realms

This list identifies from what realm traffic is coming and is used for routing by ingress realm. The source realms identified in the list must correspond to the valid realm IDs you have already configured
- Local policy attributes

The attributes serve as an expression of preference, a means of selecting one route over another. They contain information such as the next signaling address to use (next hop) or whether you want to select the next hop by codec, the realm of the next hop, and the application protocol to use when sending a message to the next hop. You can also use the attributes to filter specific types of traffic.

Routing Calls by Matching Digits

Local policy routing of a call can be based on matching a sequence of digits against what is defined in the local policy. This sequence refers to the first digits in the (phone) number, matching left to right.

The following examples show how the Net-Net system matches an area code or number code against configured local policies.

- If the number or area code being matched is 1234567 (where 123 is an area code), and the From address value in one local policy is 123, and the From address value in another local policy is 12, the Net-Net system forwards the call to the server that is defined as the next hop in the local policy with 123 as the From address value.
- If the number or area code being matched is 21234, and the From address value in one local policy is 123, and the From address value in another local policy is 12, the Net-Net system will not find a match to either local policy because the first character of the number or area code must match the first character in a From address or to address field.

The following examples show how the Net-Net system matches an area or number code against different local policies: the first one has a From address value of 12 and the second has a From address value of 123. The Net-Net system chooses the route of the local policy that is configured with the most digits matching the area or number code in its From address and To address fields.

- When the two different local policies route to two different servers, and the area or number code being matched is 123, the Net-Net system selects the second local policy based on the From address value of 123.
- When the two different local policies route to two different servers, and the area or number code being matched is 124, the Net-Net system selects the first local policy based on the From address value of 12.

Route Preference

The Net-Net 9200 builds a list of possible routes based on the source realm and the From-address (From-URI) and To-address (Request-URI), which forms a subset from which preference then decides. Any local policy routes currently outside of the configured time/day are not used, if time/day are set. Also, any local policy routes not on the list of carriers (if carriers is set and the requests has a Carrier header) are not used.

Note: The source realm is used in the `local policy` lookup process, but it is not used in route preference calculations.

Local Policy: DTMF-Style URI Routing

The Net-Net SBC supports the alphanumeric characters a-d, A-D, the asterisk (*), and the ampersand (#) for local policy matching purposes. The Net-Net SBC handles these characters as standards DN (POTS) or FQDN when found in the to-addr (req-uri username) or from-addr (from0uri username for SIP, SIPS, and TEL URIs).

In addition, before performing the lookup match, the Net-Net SBC strips characters that provide ease-of-reading separation. For example, if the Net-Net SBC were to receive a req-uri containing `tel:a-#1-781-328-5555`, it would treat it as `tel:a#17813285555`.

There are no changes to the ACLI to support this feature. However, alpha-numeric-entries must be appended with the prefix **DS:**. This stands for Dialed String, and means that your entry for a **from-address** value would be **DS:12#*34** to indicate routing based on Username 12#*34.

LP Routing with ENUM provided SA Route

You can configure the Net-Net SBC to match to Net-Net 3000 and 4000 Series behavior with regard to a session agent route as chosen by a local policy match, returned from an ENUM query, see [ENUM Lookup \(613\)](#).

When the Net-Net 9200 matches a session agent route from a local policy ENUM query, the systems forward the message toward the chosen session agent with two restrictions:

1. The Net-Net SBC does not enforce session agent constraints and collected statistics for flows between the system and the chosen session agent.
2. If no realm is configured in the matching policy attribute, the traffic is not forwarded into the session agent's realm (configured in the session agent configuration element), but to the sip-config's egress realm.

To override this behavior and configure the Net-Net 9200 to mimic Net-Net 3000 and 4000 series behavior, you must set the **enum agent match** parameter in the sip-config configuration element to **enabled**. Doing so will cause the Net-Net 9200 to:

1. Enforce session agent constraints and collected statistics for flows between the system and the chosen session agent.
2. Forward traffic to the chosen session agent's realm (in the session agent configuration element), if the chosen policy attribute's realm parameter is blank. If these first two realms are unknown (policy attribute and session agent), the Net-Net 9200 sends the traffic into the sip-config's realm.

All local policy queries that return session agent targets are matched based on the [Local Policy Session Agent Matching for SIP \(320\)](#) in the SIP Services chapter.

Upgrades

Because enabling this behavior has dramatic effects on routing behavior, this parameter's value defaults to disabled when upgrading to a release with **enum agent match** available for the first time. When creating a sip-config configuration element for the first time, the **enum agent match** parameter will be set to enabled.

SIP Routing

This section describes SIP session routing. When routing SIP call requests, the Net-Net 9200 communicates with other SIP entities, such as SIP user devices, other SIP proxies, and so on, to decide what SIP-based network resource each session should visit next. The Net-Net 9200 processes SIP call requests and forwards the requests to the destination endpoints to establish, maintain, and terminate real-time multimedia sessions.

Certain items in the messages are matched with the content of the local policy, within constraints set by the previous hop session agent, and the SIP configuration information (for example, carrier preferences) to determine a set of applicable next hop destinations.

The sending session agent is validated as either a configured session agent or a valid entry in a user cache. If the session INVITATION does not match any registering user, the SIP proxy determines the destination for routing the session INVITATION.

Limiting Route Selection Options for SIP

You can configure the local policy to use the single most-preferred route. And you can configure the SIP configuration `max_routes` option to restrict the number of routes which can be selected from a local policy lookup:

- A `max-routes=1` value limits the Net-Net 9200 to only trying the first route from the list of available preferred routes.
- A `max-routes=0` value or no `max-routes` value configured in the `options` field allows the Net-Net 9200 to use all of the routes available to it.

About Loose Routing

According to RFC 3261, a proxy is loose routing if it follows the procedures defined in the specification for processing of the Route header field. These procedures separate the destination of the request (present in the Request-URI) from the set of proxies that need to be visited along the way (present in the Route header field).

When the SIP NAT's route home proxy field is set to `enabled`, the Net-Net 9200 looks for a `session agent` that matches the `home proxy address` and checks the `loose routing` field value. If the `loose routing` is:

- `enabled`: a Route header is included in the outgoing request in accordance with RFC 3261.
- `disabled`: a Route header is not included in the outgoing request; in accordance with the route processing rules described in RFC 2543 (referred to as strict routing). That rule caused proxies to destroy the contents of the Request-URI when a Route header field was present.

Whether `loose routing` field is enabled is also checked when a `local policy's` `next hop` value matches a `session agent`. Matching occurs if the `hostname` or the `session agent's IP address` field value corresponds to the `next hop` value. If loose routing is enabled for the matching session agent, the outgoing request retains the original Request-URI and Route header with the `next hop` address.

About the Ingress Realm

You can create a list of realms in your local policy that is used by the Net-Net 9200 to determine how to route traffic. This list determines from which realm traffic is coming and is used for routing by ingress realm.

The `source realm` values must correspond to valid `identifier` entered when the realm was configured.

About the Egress Realm

An egress realm allows SIP signaling to travel out of the Net-Net system through a network other than the home realm. The Net-Net system uses egress realms for signaling purposes (when matching flows). When a packet arrives at the Net-Net system with a destination address that does not match any defined session agents, the Net-Net system uses the address associated with the `realm` that is, in turn, associated with the SIP configuration's egress realm ID, as the outgoing network. With the use of the egress realm ID, it is possible to define a default route for SIP requests addressed to destinations outside the home realm. If no egress realm is defined, the home realm (default ingress realm) is used as the default egress realm.

About SIP Redirect

SIP redirect involves proxy redirect and tunnel redirect.

Proxy Redirect

You can configure the SIP proxy mode to define how the SIP proxy will forward requests coming from the session agent. This value is used if the session agent's `proxy mode` has no value (is empty).

Tunnel Redirect

You can use `tunnel redirect` when requests are routed to a server behind a SIP NAT that sends redirect responses with addresses that should not be modified by the SIP NAT function. For example, a provider might wish to redirect certain calls (like 911) to a gateway that is local to the UA that sent the request. Since the gateway address is local to the realm of the UA, it should not be modified by the SIP NAT of the server's realm. Note that the server must have a session agent configured with the `redirect-action` field set to the `proxy` option in order to cause the redirect response to be sent back to the UA.

SIP Method Matching and To Header Use for Local Policies

For SIP, this feature grants you greater flexibility when using local policies and has two aspects: basing local policy routing decisions on one or more SIP methods you configure and enabling the Net-Net SBC to use the TO header in REGISTER messages for routing REGISTER requests.

SIP Methods for Local Policies

This feature allows the Net-Net SBC to include SIP methods in routing decisions. If you want to use this feature, you set a list of one or more SIP methods in the local policy attributes. These are the SIP methods you can enter in the list: INVITE, REGISTER, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, MESSAGE, and PUBLISH.

After the Net-Net SBC performs a local policy look-up for SIP, it then searches for local policy attributes that have the request method configured in the p. If it finds a set of policy attributes that matches a method that matches the traffic it is routing, the Net-Net SBC uses that set of policy attributes. This means that the Net-Net SBC considers first any policy attributes with methods configured before it considers those that do not have methods. In the absence of any policy attributes with methods, the Net-Net SBC uses the remaining ones for matching.

In cases where it finds neither matching policy attributes with methods or matching policy attributes without them, the Net-Net SBC either rejects the calls with a **404 No Routes Found** (if the request calls for a response) or drops the call.

You configure local policy matching with SIP methods in the local policy attributes parameter calls **methods**. This parameter is a list that takes either one or multiple values. If you want to enter multiple values, you put them in the same command line entry, enclosed in quotation marks and separated by spaces.

To configure SIP methods for local policy matching:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **local-policy** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type **policy-attributes** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy))# policy-attributes
ACMEPACKET(policy-attributes)#
```

5. **methods**—Enter the SIP methods you want to use for matching this set of policy attributes. Your list can include: INVITE, REGISTER, PRACK, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, UPDATE, MESSAGE, and PUBLISH.

By default, this parameter is empty—meaning that SIP methods will not be taken into consideration for routing based on this set of policy attributes.

If you want to enter more than one method, your entry will resemble the following example.

```
ACMEPACKET(local-policy-attributes)# methods "PRACK INFO REFER"
```

6. Save and activate your configuration.

Routing Using the TO Header

For the Net-Net SBC's global SIP configuration, you can enable the use of an ENUM query to return the SIP URI of the Registrar for a SIP REGISTER message. Without this feature enabled, the Net-Net SBC uses the REQUEST URI. This ability can be helpful because REGISTER messages only have the domain in the REQUEST URI, whereas the SIP URI in the To header contains the user's identity.

There are two parts to enabling this feature. First, you must enable the **register-use-to-for-lp** parameter in the global SIP configuration. Then you can set the next-hop in the applicable local policy attributes set to **ENUM**.

To enable your global SIP configuration for routing using the TO header:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **sip-config** and press <Enter>. If you are adding this feature to a pre-existing SIP configuration, you will need to select and edit it.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. **register-use-to-for-lp**—Set this parameter to enabled if you want the Net-Net SBC to use, for routing purposes, an ENUM query to return the SIP URI of the Registrar for a SIP REGISTER message. This parameter defaults to **disabled**.

To set up your local policy attributes for routing using the TO header:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **local-policy** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```

4. Type **policy-attributes** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy))# policy-attributes
ACMEPACKET(policy-attributes)#
```

5. **next-hop**—This is the next signaling host. Set this parameter to ENUM if you want to use SIP methods in local policy attribute information for routing purposes. You can use the following as next hops:

- IPv4 address or IPv6 address of a specific endpoint
- Hostname, IPv4 address, or IPv6 address of a configured session agent
- Group name of a configured session agent group

Note: The group name of a configured session agent group must be prefixed with SAG:

For example:

```
policy-attribute
      next-hop          SAG:appserver
policy-attribute
      next-hop          lrt:routetable
policy-attribute
      next-hop          enum:lrg
```

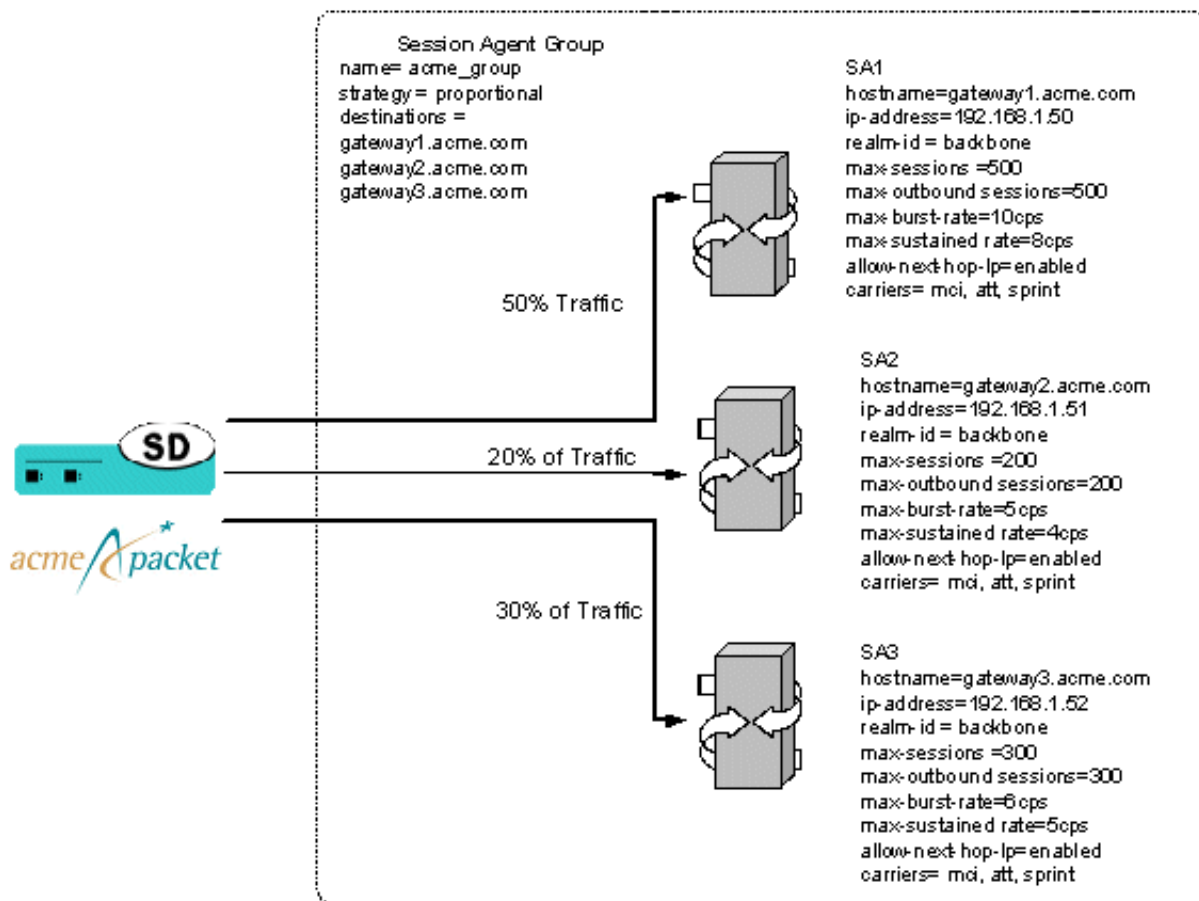
You can also configure a next hop that has an address of 0.0.0.0, thereby creating a null route. Different from not having a local policy configured (which would trigger Net-Net SBC local policy recursion), this terminates local policy recursion and immediately fails the request. In these cases, the Net-Net SBC responds with a **404 Not Found**.

6. Save and activate your configuration.

Load Balancing

This section describes Net-Net 9200 load balancing. You can use session agent groups to assist in load balancing among session agents. You define concurrent session capacity and rate attributes for each session agent and then define the session agent group. Next, you select the allocation strategy you want applied to achieve the load balancing you want.

The following example shows a configuration for load balancing gateways based on a proportional allocation strategy.



Routing and load balancing capabilities include the following:

- Least cost, which includes cost-based and time-based routing
- Customer preference
- Traffic aggregation
- Routing by media (codec) type
- Capacity-based, by destination
- Service element load balancing
- Service element failure detection and re-route
- Session agent failure
- Routing by codec

Configuring Routing

This section explains how to configure routing on the Net-Net 9200.

Configuration Prerequisite

You should have already configured the realms for your environment before you configure the routing elements. See *Configuring Realms* for details. You need to know the realm identifier when configuring session agents and local policy.

You can use an asterisk (*) when the session agent exists in multiple realms.

Configuration Order

Recommended order of configuration:

- realm
- session agent
- session agent group
- local policy

Session Agent: CLI Instructions and Examples

You can enable, then configure, individual constraints that are applied to the sessions sent to the session agent. These constraints can be used to regulate session activity with the session agent. In general, session control constraints are used for session agent groups or SIP proxies outside or at the edge of a network. Some individual constraints, such as maximum sessions and maximum outbound sessions are not applicable to core proxies because they are transaction stateful, instead of session stateful. Other constraints, such as maximum burst rate, burst rate window, maximum sustained rate, and sustained rate are applicable to core routing proxies.

To configure session agents:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **hostname**—Enter the name of the host associated with the session agent in either hostname or FQDN format, or as an IP address.

If you enter the host name as an IP address, you do not have to enter an IP address in the optional IP address parameter. If you enter the host name in FQDN format, and you want to specify an IP address, enter it in the optional IP address parameter. Otherwise you can leave the IP address parameter blank to allow a DNS query to resolve the host name.

If the initial DNS query for the session agent fails to get back any addresses, the session agent is put out-of-service. When session agent is pinged, the DNS query is repeated. The ping message is not sent until the DNS query gets back one or more IP addresses. After the query receives some addresses, the ping message is sent. The session agent remains out of service until one of the addresses responds.

Note: The value you enter here must be unique to this session agent. No two session agents can have the same hostname.

The hostnames established in the session agent populate the corresponding fields in other elements.

5. **ip-address**—*Optional*. Enter the IP address for the hostname you entered in FQDN format if you want to specify the IP address. Otherwise, you can leave this parameter blank to allow a DNS query to resolve the host name.
6. **port**—Enter the number of the port associated with this session agent. Available values include:
 - zero (0): if you enter zero (0), the Net-Net system will not initiate communication with this session agent (although it will accept calls).
 - 1025 through 65535

The default value is 5060.

Note: If the transport method value is TCP, the Net-Net system will initiate communication on that port of the session agent.

7. **state**—Enable or disable the session agent by configuring the state. By default, the session agent is enabled.
8. **app-protocol**—Enter SIP for the protocol type.
9. **transport-method**—Indicate the IP protocol to use (transport method) to communicate with the session agent. The following protocols are supported:

IP Protocol	Description
UDP	Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
UDP+TCP	Allows an initial transport method of UDP, followed by a subsequent transport method of TCP if and when a failure or timeout occurs in response to a UDP INVITE. If this transport method is selected, INVITEs are always sent through UDP as long as a response is received.
DynamicTCP	dTCP indicates that dynamic TCP connections are the transport method for this session agent. A new connection must be established for each session originating from the session agent. This connection is torn down at the end of a session.
StaticTCP	sTCP indicates that static TCP connections are the transport method for this session agent. Once a connection is established, it remains and is not torn down.

10. **realm-id**—*Optional*. Indicate the ID of the realm in which the session agent resides.

The realm ID identifies the realm for sessions coming from or going to this session agent. For requests coming from this session agent, the `realm ID` identifies the ingress realm. For requests being sent to this session agent, the `realm ID` identifies the egress realm. In a Net-Net 9200, when the ingress and egress realms are different, the media flows must be *steered* between the realms.

- No value—The egress realm is used unless the local policy dictates otherwise
- Asterisk (*)—Keep the egress realm based on the Request URI

Note: The `realm ID` you enter here must match the valid identifier value entered when you configured the realm.

11. **description**—*Optional*. Enter a descriptive name for this session agent.
12. **carriers**—*Optional*. Add the `carriers` list to restrict the set of carriers used for sessions originating from this session agent.

Carrier names are arbitrary names that can represent specific service providers or traditional PSTN telephone service providers (for sessions delivered to gateways). They are global in scope, especially if they are exchanged in TRIP. Therefore, the definition of these carriers is beyond the scope of this documentation.

You could create a list using carrier codes already defined in the North American Numbering Plan (NANP); or those defined by the local telephone number or carrier naming authority in another country.

Note: If this list is empty, any carrier is allowed. If it is not empty, only local policies that reference one or more of the carriers in this list will be applied to requests coming from this session agent.

You can use the **add** and **delete** commands when you enter the **carriers** parameter to edit the carriers list; entering a new list of values without the **add** or **delete** commands overwrites the entire list.

13. **allow-next-hop-lp**—Indicate whether this session agent can be used as a next hop in the local policy.

If you retain the default value of enabled, the session agent can be used as the next hop for the local policy. If you do not want the session agent used as the next hop, enter disabled.

14. **constraints**—Set the `constraints` parameter to enabled to indicate that the individual constraints you configure in the next step are applied to the sessions sent to the session agent. Retain the default value of disabled if you do not want to apply the individual constraints.

Note: In general, session control constraints are used for SAGs or SIP proxies outside or at the edge of a network.

15. Enter values for the individual constraints you want applied to the sessions sent to this session agent. The following table lists the available constraints along with a brief description and available values.

Constraint	Description
maximum sessions	Maximum number of sessions (inbound and outbound) allowed by the session agent. The range of values is: • minimum: zero (0) is the default value and means there is no limit • maximum: $2^{32} - 1$
maximum outbound sessions	Maximum number of simultaneous outbound sessions (outbound from the Net-Net system) that are allowed from the session agent. The range of values is: • minimum: zero (0) is the default value and means there is no limit • maximum: $2^{32} - 1$ The value you enter here cannot be larger than the maximum sessions value.
maximum burst rate	Number of session invitations allowed to be sent to or received from the session agent within the configured burst rate window value. SIP session invitations arrive at and leave from the session agent in intermittent bursts. By entering a value in this field, you can limit the amount of session invitations that are allowed to arrive at and leave from the session-agent. For example, if you enter a value of 50 here and a value of 60 (seconds) for the burst rate window constraint, no more than 50 session invitations can arrive at or leave from the session agent in that 60 second time frame (window). Within that 60-second window, any sessions over the limit of 50 are rejected. The range of values is: • minimum: zero (0) session invitations per second • maximum: $2^{32} - 1$ session invitations per second Zero is the is the default value.

Constraint	Description
maximum sustain rate	Maximum rate of session invitations (per second) allowed to be sent to or received from the session agent within the current window. The current rate is determined by counting the number of session invitations processed within a configured time period and dividing that number by the time period. By entering a value in this field, you can limit the amount of session invitations that are allowed to arrive at and leave from the session agent over a sustained period of time. For the sustained rate, the Net-Net system maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes. For example, if you enter a value of 5000 here and a value of 3600 (seconds) for the sustain rate window constraint, no more than 5000 session invitations can arrive at or leave from the session agent in any given 3600 second time frame (window). Within that 3600-second window, sessions over the 5000 limit are rejected. The range of values is: • minimum: zero (0) invitations per second • maximum: $2^{32} - 1$ invitations per second Zero is the is the default value. The value you set here must be larger than the value you set for the maximum burst rate constraint.
time to resume	Time in seconds after which the SIP proxy resumes sending session invitations to this session agent. This value only takes effect when the SIP proxy stops sending invitations because a constraint is exceeded. The range of values is: • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds Default is zero.
time to resume (ttr) no response	Delay in seconds that the SIP proxy must wait from the time that it sends an invitation to the session agent and gets no response before it tries again. The range of values is: • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds Default is zero. The value you enter here must be larger than the value you enter for the time to resume constraint.

Constraint	Description
in service period	Amount of time in seconds the session agent must be operational (once communication is re-established) before the session agent is declared as being in-service (ready to accept session invitations). This value gives the session agent adequate time to initialize. The range of values is: • minimum: zero (0) seconds • maximum: 2^{32} -1 seconds Default is zero.
burst rate window	Burst window period (in seconds) that is used to measure the burst rate. The term <i>window</i> refers to the period of time over which the burst rate is computed. Refer to the maximum burst rate information. The range of values is: • minimum: zero (0) seconds • maximum: 2^{32} -1seconds Zero is the is the default value. The value you set here must be smaller than the value you set for the maximum burst rate constraint.
sustain rate window	Sustained window period (in seconds) that is used to measure the sustained rate. Refer to the maximum sustain rate information. The range of values is: • minimum: zero (0) seconds • maximum: 2^{32} -1seconds Zero is the is the default value. The value you set here must be larger than the value you set for the maximum sustain rate constraint.

16. **req-uri-carrier-mode**—Set whether you want the selected carrier (determined by a value in the local policy) added to the outgoing message.

You can set this parameter to let the system perform simple digit translation on calls sent to gateways. A 3-digit prefix is inserted in front of the telephone number (the Request-URI) that the gateway will use to select a trunk group. Most often, the Net-Net system needs to insert the carrier code into the signaling message that it sends on.

The following table lists the available modes.

Carrier Mode	Description
None	Default value. Carrier information will not be added to the outgoing message.
URI parameter	Adds a parameter to the Request-URI. For example, cic-XXX.
Prefix	Adds the carrier code as a prefix to the telephone number in the Request-URI (in the same manner as PSTN).

17. **proxy-mode**—Indicate the proxy mode to use when a SIP request arrives from this session agent.

If this field is empty (upon initial runtime or upgrade), it's value is set to the value of the SIP configuration's `proxy mode` by default. If no proxy mode value was entered for the SIP configuration, the default for this field is set to `proxy`.

The following table lists the available proxy modes along with a brief description.

Proxy Mode	Description
Proxy	If the Net-Net system is a Session Router, the system will proxy the request coming from the session agent and maintain the session and dialog state. If the Net-Net system is a Session Director, the system behaves as a B2BUA when forwarding the request.
Redirect	The system sends a SIP 3xx reDIRECT response with contacts (found in the local policy) to the previous hop.

18. **redirect-action**—Indicate the action you want the SIP proxy to take when it receives a Redirect (3XX) response from the session agent.

If the response comes from a session agent and this field is empty (upon initial runtime or upgrade), the redirect action will be `recurse`. If no session agent is found (for example, if a message comes from an anonymous user agent), the redirect action is set to `proxy`. If the Redirect (3xx) response does not have any Contact header, the response will be sent back to the previous hop.

The following table lists the available proxy actions along with a brief description.

Proxy Action	Description
Proxy	The SIP proxy passes the response back to the previous hop; based on the proxy mode of the original request.
Recurse	The SIP proxy serially sends the original request to the list of contacts in the Contact header of the response (in the order in which the contacts are listed in the response). For example, if the first one fails, the request will be sent to the second, and so on until the request succeeds or the last contact in the Contact header has been tried.

19. **loose-routing**—Set loose routing to enabled if you want to use it (as opposed to strict routing).

When the SIP NAT route home proxy parameter is enabled, the Net-Net 9200 looks for a `session agent` that matches the home proxy address and checks the `loose routing` value. If `loose routing` is enabled, a `Route` header is included in the outgoing request in accordance with RFC 3261. If `loose routing` is disabled, the `Route` header is not included in the outgoing request (in accordance with strict routing procedures defined in RFC 2543).

The `loose routing` value is also checked when the local policy's `next hop` value matches a `session agent`. If `loose routing` is set to enabled, the outgoing request retains the original `Request-URI` and `Route` header with the next hop address.

20. **response-map**—*Optional*. Enter the name of the response map to use for this `session agent`. The mappings in each `SIP response map` is associated with a corresponding session agent.
21. **ping-method**—Indicate the SIP message/method to use to ping a session agent. The ping confirms whether the session agent is in service. If this field is left empty, no session agent will be pinged.

Setting this field value to the `OPTIONS` method might produce a lengthy response from certain session agents and could potentially cause performance degradation on your Net-Net system.

22. **ping-interval**—Indicate how often you want to ping a session agent by configuring the ping interval parameter. Enter the number of seconds you want the Net-Net system to wait between sending pings to this session agent. The default value is 32 seconds.

The Net-Net 9200 only sends the ping if no SIP transactions have occurred to/from the session agent within the time period you enter here.

23. **options**—*Optional*. You can add your own features and/or parameters by using the options parameter. You enter a comma-separated list of either or both of the following:

- `feature=value feature`

For example:

You can include the original address in the SIP message from the Net-Net 9200 to the proxy in the `Via` header parameter by entering the following option:

```
via-origin=<parameter-name>
```

The original parameter is included in the `Via` of the requests sent to the session agent. The `via origin` feature can take a value that is the parameter name to include in the `Via`. If the value is not specified for `via origin`, the parameter name is `origin`.

Note: If the feature value itself is a comma-separated list, enclose it within quotation marks.

24. **in-translationid**—*Optional*. Enter the In Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to incoming traffic.
25. **out-translationid**—*Optional*. Enter the Out Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to outgoing traffic.

Address translations attached to session agents take precedence over address translations attached to realms. If no address translation is applied to a session

agent, then the Net-Net 9200 will use the address translation applied to a realm. If an address translation is applied to both a realm and session agent, the translation attached to the session agent will apply. If the applicable session agent and realm have no associated translations, then the addresses will remain in their original forms and no address translations will be performed.

26. **trust-me**—Indicate whether this session agent is a trusted source, which the Net-Net 9200 checks when it receives a message to determine if the source is trusted.

To configure session agents:

The following example shows a session agent with an IP address used for the hostname.

```
session-agent
  hostname                192.168.1.10
  ip-address              192.168.1.10
  port                   5060
  state                  enabled
  app-protocol            SIP
  transport-method        UDP
  realm-id               realm-1
  description            englab
  carriers
    carrier1
  allow-next-hop-lp      enabled
  constraints            disabled
  max-sessions           355
  max-outbound-sessions  355
  max-burst-rate         50
  max-sustain-rate       3000
  time-to-resume         60
  ttr-no-response        0
  in-service-period      30
  burst-rate-window      60
  sustain-rate-window    3600
  req-uri-carrier-mode    None
  proxy-mode             Proxy
  redirect-action         Recurse
  loose-routing          enabled
  response-map
  ping-method
  ping-interval          0
  media-profiles
  in-translationid
  out-translationid
  trust-me               disabled
  request-uri-headers
  stop-recurse
  local-response-map
  ping-to-user-part
  ping-from-user-part
  li-trust-me            disabled
  in-manipulationid
  out-manipulationid
  p-asserted-id
```

```
trunk-group
max-register-sustain-rate    0
```

Session Agent Group: ACLI Instructions and Examples

To configure session agent groups:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-group** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-group
ACMEPACKET(session-agent-group)#
```
4. **group-name**—Enter a unique name for the session agent group in Name format.
5. **description**—*Optional*. Enter descriptive information about the session agent group.
6. **state**—Set the session agent group to **enabled** or **disabled** on the Net-Net system.
7. **application-protocol**—Set the signaling protocol to SIP.
8. **strategy**—Indicate the session agent allocation strategy you want to use. The strategy you chose selects the session agents that will be made available by this session agent group. Available values include:

Allocation Strategy	Description
hunt	Selects session agents in the order in which they are listed. For example, if the first agent is online, working, and has not exceeded defined constraints; all traffic is sent to the first agent. If the first agent is offline or if it exceeds a defined constraint, the second agent is selected. If the first and second agents are offline or exceed defined constraints, the third agent is selected. And so on through the list of session agents.
round robin	Selects each session agent in the order in which they are listed in the destination list, selecting each agent in turn, one per session.
least busy	Selects the session agent that has the fewest number of sessions relative to the maximum outbound sessions constraint or the maximum sessions constraint (for example, lowest percent busy).
proportional distribution	Based on programmed, constrained session limits, the Proportional Distribution strategy proportionally distributes the traffic among all of the available session agents.
lowest sustained rate	The Lowest Sustained Rate strategy routes to the session agent with the lowest sustained rate of session initiations/invitations (based on observed sustained session request rates).

9. **destination**—Identify the destinations (session agents) available for use by this session agent group.

The value you enter here must correspond to a valid group name for a configured session agent group or a valid hostname or IP address for a configured session agent.

The following example shows a session agent group using the SIP protocol and with the Hunt allocation strategy applied.

```
session-group
  group-name          proxy-sag1
  description         proxies for external domain
  state               enabled
  app-protocol        SIP
  strategy            Hunt
  dest
    gw-sag1
    gw-sag2
  last-modified-date  2005-01-09 23:23:36
```

SAG Matching for ENUM

When this feature is enabled and a match is found, the Net-Net SBC uses the matching SAG for routing. When there is no match for the SAG, the Net-Net SBC processes the result as it would have if this feature had not been enabled: either matching to a session agent hostname, or performing a DNS query to resolve it.

For more information, refer to this chapter's [ENUM Lookup \(613\)](#) section.

Note that you set the state of this feature in the SIP configuration.

To configure a SAG for ENUM matching:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

If you are adding support for this feature to a pre-existing SIP configuration, then you must select (using the ACLI **select** command) that configuration to edit it.
4. **enum-sag-match**—Set this parameter to **enabled** so the Net-Net SBC will match session agent group (SAG) names with the hostname portion in the naming authority pointer (NAPTR) from an ENUM query or LRT next-hop entry. The default value is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

Local Policy: ACLI Instructions and Examples

To configure local policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. **from-address**—Indicate the originating address information by entering a From address value. You can use the asterisk (*) as a wildcard to indicate this policy can be used with all originating addresses.

You can also use complete or partial E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:

 - SIP From address
 - FQDNs
 - IP addresses
5. **to-address**—Indicate the destination address by entering a To address value. You can use the asterisk (*) as a wildcard to indicate all this policy can be used for any destination address.

You can also use E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:

 - SIP Request-URI
 - FQDNs
 - IP addresses
6. **source-realm**—Enter the realm, or list of realms, you want the Net-Net 9200 to use to determine how to route traffic. This list identifies from what realm traffic is coming and is used for routing by ingress realm by the local policy.

You can use the asterisk (*) as a wildcard to indicate this local policy can be used with all realms. Or you can enter a value that corresponds to the identifier of an already configured realm. Formats include the following:

 - realm ID
 - customer name
 - peer name
 - subdomain name
 - VPN identifier
7. **activate-time**—Set the time you want the local policy to be activated using the following syntax:


```
yyyy:mm:dd hh:mm:ss
```

```
yyyy:mm:dd-hh:mm:ss
```
8. **deactivate-time**—Set the time you want the local policy to be deactivated using the following syntax:

yyyy:mm:dd hh:mm:ss

yyyy:mm:dd-hh:mm:ss

9. **state**—Set the local policy to enabled or disabled on the system.
10. **policy-attribute**—Configure local policy attributes by following the steps in the [Local Policy Attributes: ACLI Instructions and Examples \(609\)](#).
11. **description**—Enter an optional description for this configuration element. If your entry contains any spaces, you must put the entire entry in double quotation marks.

Local Policy Attributes: ACLI Instructions and Examples

To configure local policy attributes:

1. From within the local policy configuration, type **policy-attribute**. The system prompt will change to let you know that you can configure individual parameters.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```

2. **next-hop**—Identify the next signaling host by entering the next hop value. You can use the following as next hops:

- IPv4 address of a specific endpoint
- Hostname or IPv4 address of a configured session agent
- Group name of a configured session agent group

3. **realm**—Identify the egress realm (the realm used to reach the next hop) if the Net-Net 9200 must send requests out from a specific realm.

The value you enter here must correspond to a valid identifier you enter when you configured the realm. If you do not enter a value here, and the next hop is a session agent, the realm identified in the session agent configuration is used for egress.

4. **action**—Set the type of action that you want to use for this route. Valid entries are: none (default), replace-uri, and redirect.
5. **carrier**—*Optional*. Enter the name of the carrier associated with this route. The value you enter here must match one or more of the carrier names in the session agent configuration.

Entries in carrier fields can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! " # \$ % ^ & * () + - = < > ? ' | { } [] @ / \ ' ~ , . _ : ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carrier are valid carrier field formats.

6. **start-time**—Set the time of day (from the exact minute specified) the local policy attributes go into effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

1400

The default value of 0000 means that the defined policy attributes can be considered in effect any time after 00:00:00.

7. **end-time**—Indicate the time of day (from the exact minute specified) the local policy attributes are no longer in effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

2400

The default value of 2400 means that the defined policy attributes can be considered in effect any time before midnight.

8. **days-of-week**—Enter any combination of days of the week (plus holidays) you want the local policy attributes to be in effect. You must enter at least one day or holiday here. A holiday entry must correspond with a configured holiday established in the Session Router.

Available values include:

- U (Sunday)
- M (Monday)
- T (Tuesday)
- W (Wednesday)
- R (Thursday)
- F (Friday)
- S (Saturday)
- H (Holiday)

You can enter a range of values separated by a hyphen, for example U-S. And you can enter multiple values separated by commas, for example M, W, F. You cannot use spaces as separators.

9. **cost**—Enter a cost value that acts as a unitless representation of the cost of a route relative to other routes reaching the same destination (To address). This value is used as a way of ranking policy attributes.

Available values include:

- minimum: zero (0)
- maximum: 999999999

The default value is zero (0).

10. **app-protocol**—Enter the signaling protocol as SIP.
11. **state**—Indicate whether you want to enable or disable this set of local policy attributes. The default is enabled.
12. **media-profiles**—Configure a list of media profiles if you want the local policy to route SIP traffic by the codecs specified in the SDP. The list of media profiles entered here are matched against the SDP included in SIP requests and the next hop is selected by codec.

The values in this list are matched against the `rtpmap` attribute of passed SDP, and preference weight for route selection is based on the order in which the matching payload type appears in the SDP's media (`m=`) line.

For example when the following SDP arrives:

```
m=audio 1234 RTP/AVP 0 8 18
```

that contains the following attributes that correspond to three configured local policies with the same cost:

- `a=rtpmap:0 PCMU/8000`
- `a=rtpmap:8 PCMA/8000`
- `a=rtpmap:18 G729/8000`

the following route selection action occurs:

The local policy route that corresponds to the `a=rtpmap:0 PCMU/8000` attribute is selected because the payload type of 0 in the attribute line matches the first

payload type of 0 listed in the `m=` line. The codec value of PCMU indicated in this selected attribute is used to find the local policy with the media profiles attribute that includes PCMU in the list.

Because the value you enter here is matched against the codec values included in the actual passed SDP, it must correspond to accepted industry-standard codec values.

The following example shows a local policy with a next hop value of the session agent group called `gw-sag2`.

```

local-policy
  from-address
  to-address
  source-realm
  activate-time
  deactivate-time
  state
  last-modified-date
  policy-attribute
  next-hop
  realm
  replace-uri
  carrier
  start-time
  end-time
  days-of-week
  cost
  app-protocol
  state
  media-profiles

```

	*
	192.168.1.10
	*
	2005-01-20 20:30:00
	N/A
	enabled
	2005-01-10 00:36:29
	gw-sag2
	enabled
	0000
	2400
	U-S
	0
	enabled

Local Policy Matching for Parent Realms

For SIP and H.323, you can configure the Net-Net SBC to use the parent realm for routing purposes even when the source realm for an incoming message is a child realm.

With this feature disabled (default), the Net-Net SBC uses the specific source realm to perform a local policy look-up. When the source realm is a child realm and any relevant local policies are configured with the parent realm, there will be no matches and the local policy look-up will fail. To avoid this issue and ensure successful look-ups, you must configure multiple local policies if you want to use a configuration with nested realms.

The Net-Net SBC examines the source realm to determine if it is a parent realm with any child realms when you enable this feature. If the parent, source realm does have child realms, then the Net-Net SBC creates local policy entries for the parent and all of its child realms. This operation is transparent and can save time during the configuration process.

It is possible, then, for a local policy look-up to match the same child realm in two ways:

- Through a match via the parent realm
- Through a direct or non-direct match for a local policy configured with that specific child realm

In such a case, the child realm must have different costs for each type of match to avoid collisions.

This feature is enabled on a global basis in the session router configuration. Because it applies system-wide, all source realms will use this form of matching when enabled.

To enable local policy matching for parent realms:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-router** and press <Enter>.


```
ACMEPACKET(session-router)# session-router
ACMEPACKET(session-router-config)#
```
4. **match-lp-source-parent-realms**—If you want the Net-Net SBC to perform local policy realm matching based on the parent realm (so that there are local policy entries for parent and child realms), set this parameter to **enabled**. The default value is **disabled**. The valid values are:
 - enabled | disabled

```
ACMEPACKET(session-router-config)# match-lp-src-parent-realms enabled
```
5. Save and activate your configuration.

SIP Session Agent DNS-SRV Load Balancing

The Net Net SBC supports load balancing behavior as described in RFC 3263, *Session Initiation Protocol (SIP): Locating SIP Servers*.

1. The Net Net SBC will provide a new parameter **ping-all-addresses** in *session-agent* configuration mode to enable internal load balancing and RFC 3263 compliance. The Net Net SBC monitors the availability of the dynamically resolved IP addresses obtained from DNS server using OPTIONS ping (ping-per-DNS entry). The *ping-method* and *ping-interval* for each resolved IP addresses will be copied from original session-agent.

Status of Session-Agent:

In Service – if any of dynamically resolved IP addresses is in service

Out of service – if all dynamically resolved IP addresses is out of service.

The default of **ping-all-addresses** is *disabled*, in which case the Net Net SBC only pings the first available resolved IP addresses.

2. With status of each resolved IP addresses above, the Net Net SBC will recurse through the list of these in-service IP addresses dynamically resolved from DNS server on 503 response, and stop recursion based upon a configured list of response values specified by the **stop-recurse** parameter in *sip-interface* configuration mode. With internal load balancing enabled in the session-agent, the Net Net SBC provides the ability to select routing destinations based on SRV weights. The priority/weight algorithm is based on RFC 2782, *A DNS RR for specifying the location of services (DNS SRV)*.
3. The Net Net SBC will provide the similar functionality as that listed above for A-records, the SD will select first available routing destinations because there is no priority/weight contained in A-records.

ACLI Instructions and Examples

To configure the Net-Net SBC to perform Session-Agent DNS-SRV load balancing:

1. From superuser mode, use the following command sequence to access *sip-config* configuration mode. While in this mode, you configure SAG-based address resolution.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

2. Use the **ping-all-addresses** parameter to enable Session-Agent DNS-SRV load balancing.

```
ACMEPACKET(session-agent)# ping-all-addresses enabled
ACMEPACKET(session-agent)#
```

3. Use **done**, **exit**, and **verify-config** to complete Session-Agent DNS-SRV load balancing configuration.

Maintenance and Troubleshooting

The **show agents** ACLI command will now show the availability of dynamically resolved IP addresses

```
ACMEPACKET# show sip agents acme.engr.com
21:46:05-51-router
Session Agent acme.engr.com(core) [In Service] NO ACTIVITY
Session Agent acme.hxu.com(core) [In Service] NO ACTIVITY

Destination: 192.168.200.235 In Service
Destination: 192.168.200.231 In Service
...
...
```

ENUM Lookup

Telephone Number Mapping (ENUM from Telephone Number Mapping) is a suite of protocols used to unify the telephone system with the Internet by using E.164 addresses with the Domain Name System (DNS). With ENUM, an E.164 number can be expressed as a Fully Qualified Domain Name (FQDN) in a specific Internet infrastructure domain defined for this purpose (e164.arpa). E.164 numbers are globally unique, language independent identifiers for resources on Public Switched Telecommunication Networks (PSTNs). ITU-T recommendation E.164 is the international public telecommunication telephony numbering plan.

How ENUM Works

ENUM uses DNS-based architecture and protocols for mapping a complete international telephone number (for example, +1 202 123 1234) to a series of Uniform Resource Identifiers (URIs).

The protocol itself is defined in the document *E.164 number and DNS* (RFC 3761) that provides facilities to resolve E.164 telephone numbers into other resources or services on the Internet. The syntax of Uniform Resource Identifiers (URIs) is defined in RFC 2396. ENUM uses Naming Authority Pointers (NAPTR) records defined in RFC 2915 in order to identify available ways or services for contacting a specific node identified through the E.164 number.

Translating the Telephone Number

A telephone number is translated into an Internet address using the following steps:

1. The number is first stored in the following format, +1-202-555-1234. 1 is the country code for the United States, Canada, and the seventeen other countries that make up the North American Numbering Plan (NANP). The + indicates that the number is a complete, international E.164 telephone number.
2. All characters are removed except for the digits. For example, 12025551234.
3. The order of the digits is reversed. For example, 43215552021. The telephone number is reversed because DNS reads addresses from right to left, from the most significant to the least significant character. Dots are placed between each digit. Example: 4.3.2.1.5.5.5.2.0.2.1. In DNS terms, each digit becomes a zone. Authority can be delegated to any point within the number.
4. A domain (for example, e164.arpa) is appended to the end of the numbers in order to create a FQDN. For example, 4.3.2.1.5.5.5.2.0.2.1.e164.arpa.
5. The domain name is queried for the resource records that define URIs necessary to access SIP-based VoIP.

Once the authoritative name server for that domain name is found, ENUM retrieves relevant records and uses that data to complete the call or service. For example, the number 12025551234 returns sip:my.name@bigcompany.com.

About NAPTR Records

ENUM uses NAPTR records for URI resource records. NAPTR records are used to translate E.164 addresses to SIP addresses. An example of a NAPTR record is:

```
$ORIGIN 4.3.2.1.5.5.2.0.2.1.e164.arpa.
IN NAPTR 100 10 "u" "sip+E2U" "!^.*$!sip:phoneme@example.net!"
```

This example specifies that if you want to use the "sip+E2U" service, you should use sip:phoneme@example.net as the address.

The regular expression can be used by a telephone company to easily assign addresses to all of its clients. For example, if your number is +15554242, your SIP address is sip:4242@555telco.example.net; if your number is +15551234, your SIP address is sip:1234@555telco.example.net.

About the Net-Net SBC ENUM Functionality

The ENUM functionality lets the Net-Net SBC make an ENUM query for a SIP request. The ENUM lookup capability lets the Net-Net SBC transform E.164 numbers to URIs during the process of routing (or redirecting) a call. During the routing of a SIP call, the Net-Net SBC uses a local policy attribute to determine if an ENUM query is required and if so which ENUM server(s) need to be queried. A successful ENUM query results in a URI that is used to continue routing or redirecting the call.

**Configurable
Lookup Length**

You can configure a lookup length in the ENUM configuration that provides for more efficient caching of URI lookup results; in it, you can specify the length of the string for the DNS request starting from the most significant digit. This provides more flexibility for length matching, which is useful given the amount of wild card matching available in ENUM services. Specific ENUM groups might only be intended to provide NPANXX or wild card results.

**Custom ENUM
Service Type
Support**

You can configure the ENUM service type that you want to use for an ENUM group. While the Net-Net SBC has always supported **E2U+sip** and **sip+E2U** by default, and still does, you can configure the service type to those supported in RFCs 2916 and 3721.

For example, you can set the service type in the ENUM configuration to support **E2U+sip** and **E2U+voicemsg:sip**. When you configure customer ENUM service types on your system, however, you should note the following:

- New entries in the **service-type** parameter overwrite pre-existing values, including the default values.
- Because of the overwriting noted above, you must include the defaults (if you want them configured) when you are adding additional ENUM service type support. That is, you have to also type in **E2U+sip** and **sip+E2U** if you want them to be used in addition to the customized types you are setting.

ENUM Failover and Query Distribution

**ENUM Query
Distribution**

The Net-Net SBC can intelligently distribute ENUM queries among all configured ENUM servers. By setting the enum config's **query method** parameter to round robin, the Net-Net SBC will cycle ENUM queries, sequentially, among all configured

ENUM servers. For example, query 1 will be directed to server 1, query 2 will be directed to server 2, query 3 will be directed to server 3, and so on.

The default query method, hunt, directs all ENUM queries toward the first configured ENUM server. If the first server is unreachable, the Net-Net SBC directs all ENUM queries toward the next configured ENUM server, and so on.

Failover to New enum-config

When an enum-config's configured servers are unreachable via the network, i.e., no response is received on a query, the Net-Net SBC can failover to a defined ENUM config that contains different enum servers to query. This failover behavior works when all servers in an enum config are unreachable, rather than when the Net-Net SBC receives not-found type responses.

The Net-Net SBC queries each ENUM server once before trying the next configured server, and then ultimately trying the servers listed in the **failover-to** enum config. If the failover-to servers also are unreachable, the Net-Net SBC fails the call; the failover-to behavior does not recurse among enum-configs, it only checks the first, linked enum-config.

ENUM Server Operation States

After 5 consecutive failed attempts, an ENUM server is considered Out of Service (OOS). All subsequent queries which would be directed to the OOS servers are immediately directed to the first non-OOS server. ENUM servers return to in-service after 600 seconds. If all configured ENUM servers are OOS, the Net-Net SBC fails the call.

After the first failed attempt to reach an ENUM server, it is placed in a Time Out state, which it stays in for 30 seconds. Within this 30 seconds it will not be contacted when an ENUM query is made. After the 30 seconds pass, the ENUM server goes back to an in-service state.

Server Availability Monitoring

The Net-Net SBC can probe an ENUM server's health by sending it a standard ENUM NAPTR query and receiving a valid answer. The query is for the phone number defined in the **health query number** parameter, which should be one that the ENUM servers can positively resolve. As long as the query succeeds, that ENUM server maintains its in-service state and is available for ENUM queries. Any lack of response, whether network based (time-outs), or application based (DNS error or "not found" response) is considered a query failure and the server is set to OOS and unavailable for ENUM queries.

The Net-Net SBC continuously checks the health of all configured ENUM servers to determine their current state and monitor for failed servers' return to service. All servers are checked for availability at the **health query interval** parameter, as defined in seconds.

Note: When ENUM server availability monitoring is enabled, ENUM servers can only exist in an in-service or out-of-service states; Without the health query interval defined, server availability monitoring is disabled, and ENUM servers exist in three service states, as described in the ENUM Server Operation States section above.

ENUM Server IP Address and Port

You can configure an IP address and port for each enum server listed in the enum-servers parameter. IP address and port are specified in XXX.XXX.XXX.XXX:YYYY format with a port value range of 1024-65535. If the port number is not specified, 53 is assumed.

Caching ENUM Responses

As DNS responses often lead to further DNS queries, a DNS server can send additional multiple records in a response to attempt to anticipate the need for additional queries. The Net-Net SBC can locally cache additional NAPTR, SRV, and A records returned from an ENUM query to eliminate the need for unnecessary external DNS requests. These cached records can then be accessed by internal ENUM and DNS agents.

The unprompted NAPTR, SRV, or A record returned to the Net-Net SBC must include complete information to resolve a call to be added to the local DNS/ENUM cache, otherwise the Net-Net SBC will preform an external query to find the address it is looking to resolve.

Cached entries are per ENUM config. That means if one ENUM config has a number of cached entries, and an ENUM request is directed through a different ENUM config, the second configuration is not privy to what the first configuration has cached.

The Net-Net SBC uses the shorter lifetime of the DNS response's TTL or the server dns attribute's transaction-timeout to determine when to purge a DNS record from the local cache.

Source URI Information in ENUM Requests

ENUM queries can include the source URI which caused the ENUM request. The Net-Net SBC can add the P-Asserted-ID URI (only if not in an INVITE) or the From URI into an OPT-RR Additional Record to be sent to the ENUM server. It can be useful to specify the originating SIP or TEL URI from a SIP request which triggered the ENUM query, so the ENUM server can provide a customized response based on the caller.

This feature implements the functionality described in the Internet Draft, *DNS Extension for ENUM Source-URI*, draft-kaplan-enum-source-uri-00.

When a P-Asserted-ID is blocked or removed before the ENUM query is made, the Net-Net SBC only sends the URI in the From header.

Note that to support this feature, according to the Internet draft, ENUM clients must support 1220 bytes in UDP responses. Therefore, if this feature is enabled, and the max response size parameter is not set i.e., with a 512 byte default, the Net-Net SBC will set the size to 1200 on the OPT-RR records sent.

Operation Modes

There are four modes of ENUM operation that are selected on a global basis:

- stateless proxy
- transaction stateful proxy
- session stateful proxy
- B2BUA with or without media

Stateless Proxy Mode

The stateless proxy mode is the most basic form of SIP operation. The stateless proxy mode:

- Has the least number of messages per call. No record route header is added and there are no 100 Trying or BYEs.
- Does not keep transaction state (timers and retransmission). There are no session counters and no session stop time. No session stop time means no RADIUS STOP records.
- Has no limits on session state.
- Acts primarily as a routing device, with local policy routing and ENUM routing.

Transaction Stateful Proxy

In the transaction stateful proxy mode:

- Adds state to the proxy (not dialogs).
- Has lower number of messages per call. No Record Route header added and no BYES.
- Keeps transaction state (timers and retransmissions).
- Enforces session restrictions (32k) because of state management. These restrictions can be increased.
- Acts as routing device with transaction timers, with local policy routing and ENUM routing.
- Can off-load some transactions across unreliable links.

Session Stateful Proxy

The session stateful proxy mode:

- Maintains dialog state as a proxy.
- Includes BYES (though cannot be inserted)
- Keeps transaction state (timers and retransmission)
- Enforces session restrictions (32k) because of state management.
- Routes full sessions with transaction timers with local policy routing and ENUM routing.

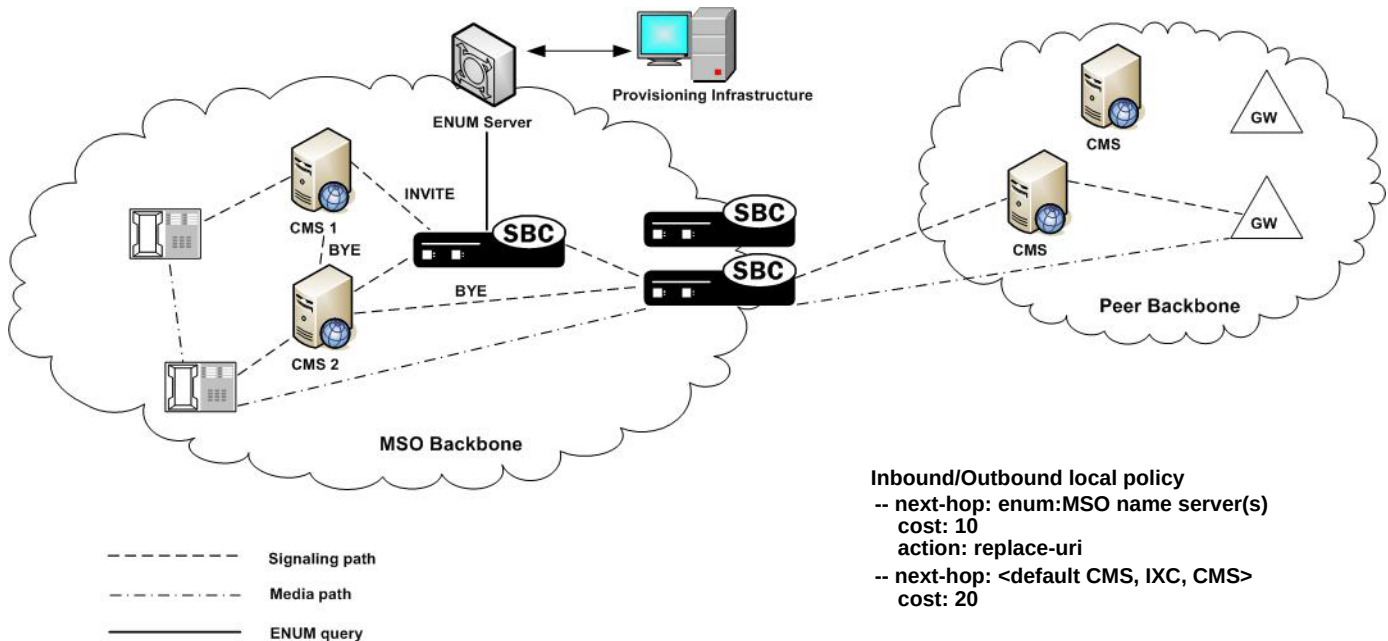
B2BUA

The B2BUA mode:

- Acts as UAS and UAC within call flow.
- Includes BYES (can be inserted).
- Keeps transaction state (timers and retransmissions)
- Enforces session restrictions (32k) because of state management.
- Routes full sessions with topology masking. Includes rewriting Via, Route, Contact headers, full NATing with SIP NAT or header manipulation, direct bridging, local policy routing, and ENUM routing.

**Example: ENUM
Stateless Proxy**

The following diagram shows the Net-Net SBC using ENUM to query a local subscriber database. The Net-Net SBC serves as the inbound and outbound routing hub and performs media management. Calls are routed throughout the MSO network using ENUM lookup results.

**ACLI Instructions
and Examples**

This section shows you how to configure ENUM on your Net-Net SBC.

**ACLI Instructions and
Examples**

To configure ENUM:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **enum-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```
4. **name**—Enter a string that uniquely identifies this ENUM configuration. You use this name in other areas of the Net-Net SBC configuration to refer to this ENUM configuration. For example, in the local policy attributes.
5. **top-level-domain**—Enter the domain extension to be used when querying the ENUM servers for this configuration. For example, e164.arpa. The query name is a concatenation of the number and the domain.

For example the number is +17813334444 and the domain is e164.arpa, the query name would be 4.4.4.4.3.3.3.1.8.7.1.e164.arpa.com.

6. **realm-id**—Enter the realm where the ENUM servers can be reached. The realm ID is used to determine on which network interface to issue the ENUM query.
7. **enum-servers**—Enter the list of ENUM servers (an ENUM server and corresponding redundant servers) to be queried. Separate each server address with a space and enclose list within parentheses.

The first server on this list is the first one to be queried. If the query times out (including retransmissions) without getting a response, the next server on the list is queried and so on.

8. **timeout**—Enter the total time in seconds that should elapse before a query sent to a server (and its retransmissions) will timeout. If the first query times out, the next server is queried and the same timeout is applied. This process continues until all the servers in the list have timed out or until one of the servers responds.

The retransmission of ENUM queries is controlled by three timers. These timers are derived from this timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the Net-Net SBC should retransmit 3 times before timing out to give the server a chance to respond.

- Init-timer is the initial retransmission interval. If a response to a query is not received within this interval, the query is retransmitted. To safeguard from performance degradation, the minimum value allowed for this timer is 250 milliseconds.
- Max-timer is the maximum retransmission interval. The interval is doubled after every retransmission. If the resulting retransmission interval is greater than the value of max-timer, it is set to the max-timer value.
- Expire-timer: is the query expiration timer. If a response is not received for a query and its retransmissions within this interval, the server will be considered non-responsive and the next server in the list will be tried.

The following examples show different timeout values and the corresponding timers derived from them.

timeout >= 3 seconds

```
Init-timer = Timeout/11
Max-Timer = 4 * Init-timer
Expire-Timer = Timeout
```

timeout = 1 second

```
Init-Timer = 250 ms
Max-Timer = 250 ms
Expire-Timer = 1 sec
```

timeout = 2 seconds

```
Init-Timer = 250 ms
Max-Timer = 650 ms
Expire-Timer = 2sec
```

9. **cache-inactivity-timer**—Enter the time interval in seconds after which you want cache entries created by ENUM requests deleted, if inactive for this interval. If the cache entry gets a hit, the timer restarts and the algorithm is continued until the cache entry reaches its actual time to live.

Setting this value to zero disables caching. For optimal performance, set this to one hour. Rarely used cache entries are purged and frequently used entries are retained.

10. **lookup-length**—Specify the length of the ENUM query, starting from the most significant digit. The valid range for this value is 1-255; the default is 0 which matches for all digits.
11. **max-response-size**—Specify the EDNS0 capability for UDP ENUM responses larger than 512 bytes. The valid range for this value is 512-65535; the default is 512.
12. **service-type**—Enter the ENUM service types you want supported in this ENUM configuration. Possible entries are **E2U+sip** and **sip+E2U** (the default), and the types outlines in RFCs 2916 and 3721.
This parameter defaults to the following service types: **E2U+sip** and **sip+E2U**.
You can enter multiple services types in the same entry, as in this example:
ACMEPACKET(enum-config)# service-type E2U+sip,sip+E2U,E2U+voicemail
13. **query-method**—Set the strategy the Net-Net SBC uses to contact ENUM servers. Valid values are:
 - **hunt**—Directs all ENUM queries toward the first configured ENUM server. If the first server is unreachable, the Net-Net SBC directs all ENUM queries toward the next configured ENUM server, and so on.
 - **round-robin**—Cycles all ENUM queries, sequentially, among all configured in-service ENUM servers. Query 1 will be directed to server 1, query 2 will be directed to server 2, query 3 will be directed to server 3.
14. **health-query-number**—Set this parameter to a standard ENUM NAPTR query that will consistently return a positive response from the ENUM server.
15. **health-query-interval**—Set this parameter to the number of seconds to perpetually probe ENUM servers for health.
16. **failover-to**—Set this parameter to the name of another ENUM-config which to failover to under appropriate conditions.
17. **cache-addl-records**—Set this parameter to **enabled** for the Net-Net SBC to add additional records received in an ENUM query to the local DNS cache.
18. **include-source-info**—Set this parameter to **enabled** for the Net-Net SBC to send source URI information to the ENUM server with any ENUM queries.
- 19.

Configuring the Local Policy Attribute

You can specify that an ENUM query needs to be done for the routing of SIP calls. You do so by configuring the local policy's next-hop attribute with the name of a specific ENUM configuration, prefixed with the **enum:** tag. For example: **enum:test**

You can configure multiple next-hops with different ENUM servers or server groups (possibly with different top-level-domains). If the first ENUM server group you enter as the next hop is not available, one of the others can be used.

To configure local policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```

3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. Type **policy-attributes** and press <Enter>. The system prompt changes to let you know you can begin configuring policy attributes.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(policy-attributes)#
```
5. **next-hop**—Enter the name of the ENUM configuration with the prefix enum:. For example, **enum:test**.
6. **action**—Set to redirect if you want to send a redirect message back to the calling party with the information returned by ENUM in the Contact. The calling party then needs to send a redirect using that information.
7. Save and activate your configuration.

Local Policy Example

The following example shows one local policy with the next-hop configured to use enum:test and a second with the next-hop configured to use enum:test with different costs.

```
local-policy
  from-address          172.16.0.11
  to-address            *
  source-realm          public
  activate-time         N/A
  deactivate-time       N/A
  state                 enabled
  policy-attributes
    next-hop            enum:test
    realm               public
    action              none
    terminate-recursion disabled
    carrier
    start-time          0000
    end-time            2400
    days-of-week        U-S
    cost                0
    app-protocol        SIP
    state               enabled
    media-profiles
```

CNAM Subtype Support for ENUM Queries

CNAM, calling name, data is a string up to 15 ASCII characters of information associated with a specific calling party name. The Internet-draft, draft-ietf-enum-cnam-08.txt, registers the ENUM service 'pstndata' and subtype 'cnam' using the URI scheme 'pstndata:' to specify the return of CNAM data in ENUM responses. The Net-Net SBC recognizes CNAM data returned via this mechanism. CNAM data is then inserted into the display name of the From: header in the original Request. If a P-Asserted-ID header is present in the original request, the CNAM data is inserted there as well.

CNAM data is identified by an ENUM response with service-type:
E2U+pstndata:cnam

CNAM support is configured in the sip profile configuration element, which can then be applied to either a session agent, realm, or SIP interface.

The Net-Net SBC can preform CNAM queries on the signaling message's ingress or egress from the system by setting the cnam lookup direction parameter to either ingress or egress. If the CNAM lookup direction parameters are configured on both the ingress and egress sides of a call, the Net-Net SBC will only preform the lookup on the ingress side of the call.

CNAM Unavailable Response

A CNAM response can include a Calling Name Privacy Indicator parameter ('unavailable=p') or Calling Name Status Indicator parameter ('unavailable=u') in responses. The Net-Net SBC can insert a custom reason string into the SIP message's From and P-Asserted-ID header in the original requires.

Configuring the **cnam unavailable ptype** parameter inserts the specified text into the From and P-Asserted-ID headers when a CNAM response contains the unavailable=p parameter.

Configuring the **cnam unavailable utype** parameter inserts the specified text into the From and P-Asserted-ID headers when a CNAM response contains the unavailable=u parameter.

SIP Profile Inheritance

CNAM features, via the SIP Profile configuration element can be applied to session agents, realms, and SIP interfaces. The more generalized object inherits the more specific object's values. For example, if CNAM support via a SIP profile is configured on a session agent, the expected processing will override any SIP profile configuration on the downstream realm or SIP interface. Likewise, if CNAM support is unconfigured on the receiving session agent, but configured in the realm, CNAM configuration on the SIP interface will be ignored.

ACLI Instructions and Examples

To enable the Net-Net SBC to preform CNAM subtype ENUM queries, you must configure a SIP profile with an enum-config object (that points to valid ENUM servers). The referenced enum-config configuration element lists the servers to contact for CNAM type queries (and other general ENUM server interaction parameters).

To configure CNAM subtype support:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```
4. **name**—Enter a string that uniquely identifies this SIP profile configuration. You use this name in other areas of the Net-Net SBC configuration to refer to this SIP profile in session agents, realms, or SIP interfaces.

5. **cnam-lookup-server**—Set this parameter to the name of an ENUM-config to that will query ENUM servers for CNAM data.
6. **cnam-lookup-dir**—Set this parameter to **ingress** or **egress** to identify where the Net-Net SBC performs a CNAM lookup with respect to where the call traverses the system. The default value is **egress**.
7. **cnam-unavailable-ptype**—Set this parameter to a string, no more than 15 characters, to indicate that the unavailable=p parameter was returned in a CNAM response.
8. **cnam-unavailable-utype**—Set this parameter to a string, no more than 15 characters, to indicate that the unavailable=u parameter was returned in a CNAM response.
9. Save your work.

Local Route Tables

Adding onto the Net-Net SBC's existing routing capabilities, the Local Route Tables feature gives the Net-Net SBC the ability to determine next hops and map E.164 to SIP URIs locally. This ability provides extensive flexibility for routing.

This feature introduces the concept of a local route cache, which is populated by a local gzipped XML file on the Net-Net SBC. Each table/cache is populated from one defined XML file. For routing, the local route cache operates in a way similar to the ENUM model in that a local policy next hop specifies the local route table that the Net-Net SBC looks up. For example, you can configure one next hop to use one table, and another next hop to use a different one.

Similar to the ENUM model, the Net-Net SBC typically performs a local route table lookup using the telephone number (TN) of the SIP Request-URI. This is the user portion of the URI, and the Net-Net SBC ignores user parameters or non-digit characters. The local route table XML file defines the matching number and the resulting regular expression replacement value—as ENUM NAPTR entries would. The Net-Net SBC uses the resulting regular expression to replace the Request-URI, and it uses the hostname or IP address portion to determine the next hop. If the hostname or IP address matches a configured session agent, the request is sent to that session agent. If the Net-Net SBC does not find a matching session agent for the hostname/IP address, it either performs a DNS query on the hostname to determine its IP address or sends the request directly to the IP address.

When the next hop is defined as a user-parameter lookup key, such as a routing number (RN) or carrier identification code (CIC), the defined key is used for the local route table lookup. For details, refer to the [Routing-based RN and CIC \(633\)](#) section of this document.

Multiple (up to 10) next hops per LRT entry are tried in the order in which they appear in the XML file. If the chosen next hop fails (for example, because it matches an out-of-service session agent or the next hop responds with a failure response), then the Net-Net SBC will try the next in the ordered list.

Note: Entering XML comments on the same line as LRT XML data is not currently supported.

How It Works

The Net-Net SBC supports a new system task for local route tables, and can perform local route table lookups for SIP requests. It is also responsible for communicating

the results to the SIP task. The new task processes the new local routing configuration objects.

When a SIP call is being routed, the Net-Net SBC uses local policy attributes to determine if a local route table lookup is required. If one is needed, it also selects which local routing configuration to use. Successful local route table lookups result in URIs that can be used to continue routing or redirecting calls.

HA Provisioning

When using the LRT feature, you must upload the local routes XML file to both standby and active SPUs; this file is not automatically replicated.

ACLI Instructions and Examples

This section shows you how to:

- Set up local route configuration
- Specify that a set of local policy attributes needs to use local routing

Setting Up a Local Routing Configuration

The local routing configuration is a new element in the ACLI session-router path. This is where you configure a name for the local route table, the filename you want to give to the database corresponding to this table, and the prefix length (significant digits/bits) to be used for lookup.

To set up a local routing configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **local-routing-config** and press <Enter>.


```
ACMEPACKET(session-router)# local-routing-config
ACMEPACKET(local-routing-config)#
```
4. **name**—Enter the name (a unique identifier) for the local route table; this name is used for reference in the local policy attributes when to specify that local routing should be used. There is no default for this parameter, and it is required.
5. **file-name**—Enter the name for the file from which the database corresponding to this local route table will be created. You should use the **.gz** format, and the file should be placed in the **/code/lrt/** directory. There is no default for this parameter and it is required.
6. **prefix-length**—Enter the number of significant digits/bits to used for lookup and cache storage. The default is **0**.
7. Save and activate your configuration.

Your configured local routing configuration will resemble the following sample.

```
local-routing-config
  name                lookup
  file-name            abc.xml.gz
  prefix-length        3
```

Applying the Local Routing Configuration

You apply the local routing configuration by calling it to use in the local policy attributes. You do this by setting a flag in the **next-hop** parameter along with the name of the local routing configuration that you want to use.

To apply the local routing configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **local-policy** and press <Enter>.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. Type **policy-attributes** and press <Enter>.


```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```
5. **next-hop**—In the **next-hop** parameter, type in **lrt:** followed directly by the name of the local routing configuration to be used. The **lrt:** tag tells the Net-Net SBC that a local route table will be used.


```
ACMEPACKET(local-policy-attributes)# next-hop lrt:lookup
```
6. Save and activate your configuration.

Local Route Table Support for H.323 and IWF

Local Route Table (LRT) support for H.323 and IWF is compatible with that currently offered for SIP. LRT and ENUM provide the Net-Net SBC with the ability to perform routing based on ENUM queries to a DNS server or local to an onboard database.

For the LRT feature, this means that entries in the local routing table now include those prefixed with the **h323:** URI scheme, indicating that H.323 is the next hop protocol.

IWF Considerations

When the system performs a local policy lookup for an incoming SIP or H.323 call and determines an ENUM/LRT server is the next hop, it queries that ENUM/LRT server. The response will include the URI scheme, indicating the next hop protocol and the hostname/IP address representing the next hop. For cases where the incoming call signaling protocol and the URI scheme of the ENUM/LRT response are the same, the call requires no interworking. The Net-Net SBC can simply route the egress call leg to the specified next hop.

However, interworking is required when the incoming signaling protocol and the URI scheme of the ENUM/LRT response do not match. In these cases, the Net-Net SBC interworks between SIP and H.323, routing the call to the appropriate next hop.

In addition, the Net-Net SBC compares the URI scheme returned in the ENUM/LRT response to the application protocol specified in the policy attributes. If the URI scheme is SIP but the policy attributes indicate H.323, the route is deemed invalid. The same is true for an H.323 URI scheme and SIP route.

ACLI Instructions and Examples

In order for LRT to work for H.323 and IWF calls, you do not have to perform any special configuration. However, you can configure the system to match ENUM/LRT responses against session agent groups, and then use those SAGs for routing.

To enable matching ENUM/LRT responses for H.323 SAG routing:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **h323-config** and press <Enter>.


```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```
4. **enum-sag-match**—Set this parameter to enabled if you want the Net-Net SBC to perform matching against the hostnames in ENUM/LRT lookup responses and session agent groups. If there is a match, the Net-Net SBC uses the matching SAG for routing. If no match is found, normal ENUM/LRT routing proceeds.

ENUM: Large UDP Datagram Support for DNS NAPTR Responses

The Net-Net SBC supports ENUM queries that manage large UDP DNS responses set out in RFC 2671, “Extension Mechanisms for DNS (EDNS0),” meaning that it can manage responses beyond its previous 512 byte limitation.

How It Works

According to RFC 2671, senders can advertise their capabilities using a new resource record (OPT pseudo-RR), which contains the UDP payload size the sender can receive. You can configure the Net-Net SBC to include this resource record in its DNS queries with ENUM.

Note, however, that if your configuration does not specify a maximum response size over 512 bytes, then the Net-Net SBC will not add the OPT pseudo-RR to the ENUM query—without which the ENUM server will truncate the response.

ACLI Instructions and Examples

This section shows you how to enable your Net-Net SBC to support UDP datagrams in excess of 512 bytes.

To enable large UDP datagram support when handling ENUM queries:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **enum-config** and press <Enter>.


```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```
4. **max-response-size**—Set the maximum size in bytes for UDP datagram responses. This parameter takes values from 512 (default) to 65535. Although the maximum value you can set is 65535, Acme Packet recommends configuring values that do not exceed 4096 bytes.

5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

Multistage Local Policy Routing

Multistage local policy routing enables the Net-Net SBC to perform multiple stages of route lookups where the result from one stage is used as the lookup key for the next routing stage.

Routing Stages

A routing stage signifies a re-evaluation of local policy based on the results of a local policy lookup. In the simplest, single stage case, the Net-Net SBC performs a local policy lookup on a SIP message's Request URI. The result of that local policy lookup is a next hop FQDN, IP address, ENUM lookup, or LRT lookup; that result is where the Net-Net SBC forwards the message. In the multistage routing model, that resultant next hop is used as the lookup key for a second local policy lookup.

The results of each stage do not erase the results of the previous stage. Thus, previous results are also possible routes to use for recursion, but the next stage results are tried first.

Note: Setting a next hop to a SAG in a multistage scenario constitutes an error.

Network Applications

The following are typical applications of multistage routing:

- An operator might need to query an ENUM server for a destination number. Based on the NAPTR result of the ENUM query, the Net-Net SBC performs a local policy lookup to decide how to route the request, perhaps based on a LRT table lookup.
- An operator might need to query one ENUM server for a number portability lookup, then based on the routing number perform a second ENUM query to a different server to learn which carrier to use for the routing number. Then, then based on the identified carrier perform a LRT lookup for what next-hop(s) to use for that carrier.
- An operator might query an LRT table to confirm the allowed source number. Then, based on the result, query an ENUM server for destination routing.

Multistage Routing Conceptual Example

Multistage routing is enabled by setting a policy attribute's lookup parameter to multi. Instead of replacing the SIP message's request URI with the policy attribute's next hop address or response from an ENUM or LRT lookup, the system uses that next hop or ENUM or LRT lookup response to reconstruct the SIP message. The reconstructed SIP message is fed again through all configured local policy configuration elements (and policy attribute sub elements). Each time the Net-Net SBC re-evaluates a SIP message against local policies, it is considered an additional routing stage. When multiple records are returned from an ENUM or LRT lookup, the Net-Net SBC evaluates the first response against all applicable local policies. If unsuccessful, the Net-Net SBC evaluates all additional responses, in turn, against all applicable local policies.

For example:



Multistage Routing Example 2

The following three local policy configuration elements are configured in the Net-Net SBC:

Local Policy 1				Local Policy 2				Local Policy 3			
from-address	*			from-address	*			from-address	*		
to-address	159			to-address	192.168.1.49			to-address	215680000002		
source-realm	private			source-realm	private			source-realm	private		
policy-attribute				policy-attribute				policy-attribute			
next-hop	lrt:default-lrt			next-hop	lrt:carrier-lrt			next-hop	192.168.200.98		
lookup	multi			lookup	multi			lookup	single		
policy-attribute				policy-attribute				policy-attribute			
next-hop	192.168.200.50			next-hop	lrt:emergency			next-hop	192.168.200.97		
lookup	single			lookup	single			lookup	single		
								policy-attribute			
								next-hop	192.168.200.44		
								lookup	multi		

The local route table in default-lrt appears as follows:

```
<route>
  <user type="E164">159</user>
  <next type="regex">!^.*$!sip:11568000000@192.168.200.47!</next>
  <next type="regex">!^.*$!sip:215680000002@192.168.200.99!</next>
  <next type="regex">!^.*$!sip:11578000000@192.168.200.44!</next>
</route>
```

1. The Net-Net SBC receives an INVITE on realm, private (SDP is omitted below):

```
INVITE sip:159@192.168.1.49:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.48:5060
From: sipp <sip:sipp@192.168.1.48:5060>;tag=1
To: sut <sip:159@192.168.1.49:5060>
Call-ID: 1-4576@192.168.1.48
CSeq: 1 INVITE
Contact: sip:sipp@192.168.1.48:5060
Max-Forwards: 70
Subject: Performance Test
```

Content-Type: application/sdp

Content-Length: 135

2. The Net-Net SBC performs a local policy search based on the following parameters:

from-address: sipp <sip:sipp@192.168.1.48:5060>;tag=1

to-address: sip:159@192.168.1.49:5060

Source Realm: private

3. The local policy search returns the four following routes to try:

lrt:default-lrt

192.168.200.50

lrt:emergency

lrt:carrier-lrt

a. The first next-hop route will be an LRT query. In addition, this policy attribute is configured with lookup=multi, meaning the results of the LRT query should be used for another local policy query, i.e., a second stage. More specifically, the request-uri that was received in response to the LRT query will be used as the to-uri in the next LP query.

b. The Net-Net SBC performs the LRT lookup in the default-lrt configuration element and is returned the following:

sip:11568000000@192.168.200.47

sip:215680000002@192.168.200.99

sip:11578000000@192.168.200.44

c. The Net-Net SBC attempts to use the results from the LRT query for the next stage Local Policy lookup(s). Beginning with the first route and continuing in sequential order, the Net-Net SBC will try to route the outgoing INVITE message by performing additional Local Policy lookups on the remaining LRT query results, until the INVITE is successfully forwarded.

The Net-Net SBC performs a local policy query on:

sip:11568000000@192.168.200.47

Which equates to a local policy lookup on:

from-URI=sipp <sip:sipp@192.168.1.48:5060>;

to-URI=sip:11568000000@192.168.200.47

Source Realm: private

The query fails because there is no Local Policy entry for 11568000000.

d. The Net-Net SBC performs a second query on request-uri

sip:215680000002@192.168.200.99

Which equates to a local policy lookup on:

from-URI=sipp <sip:sipp@192.168.1.48:5060>;

to-URI=sip:215680000002@192.168.200.99

Source Realm: private

e. The LP query is successful and returns the following next- hops:

192.168.200.98

192.168.200.99

192.168.200.44

f. The three routes shown above represent the next stage of the multistage routing for this INVITE. The policy attributes' lookup parameter is set to single for these next-hops. Therefore, the SD will attempt to send the outgoing INVITE message to one or more of these next-hops; there are no more stages to check.

4. The Net-Net SBC sends an INVITE to 192.168.200.98:


```
INVITE sip:215680000002@192.168.200.98;lr SIP/2.0
Via: SIP/2.0/UDP 192.168.200.49:5060
From: sipp <sip:sipp@192.168.1.48:5060>
To: sut <sip:159@192.168.1.49:5060>
Call-ID: SDnhae701-76e8c8b6e168958e385365657faab5cb-v3000i1
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.200.49:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
```
5. If the INVITE is sent to 192.168.200.98 successfully, the local policy routing will conclude and the call will continue processing. Otherwise the SD will try the other next hops until a route succeeds or all next-hops have been exhausted

Customizing Lookup Keys

When the **next hop** parameter points to perform an ENUM or LRT lookup, it can be provisioned with a "key=" attribute in order to specify a parameter other than the username to perform the lookup on. The following table lists the header, key value, and corresponding syntax to configure the Net-Net SBC with.

Username from Header:	Key Value	Example
To-URI	\$TO	key=\$TO
From-URI	\$FROM	key=\$FROM
P-Asserted-Identity	\$PAI	key=\$PAI

For a subsequent stage in multistage local policy routing, the lookup key to use for the next stage can be explicitly specified by configuring the **next key** parameter. By default, multistage lookups use the modified Request-URI returned from the ENUM/LRT response as the to-address key for the next local policy lookup. When the **next key** parameter is configured, its value will be used for the to-address key in the subsequent local policy lookup regardless if an ENUM or LRT lookup is configured for that policy attribute. The key syntax for this parameter is the same as with the Routing-based RN and CIC feature.

Multistage Routing Lookup Termination

It is important for the Net-Net SBC to have a mechanism to stop performing additional stages of route lookups and limit the number of attempts and results to be tried. Routing termination can be performed at in the non-multistage way or at the global session router level.

Global Local Policy Termination

The Net-Net SBC can be configured to limit local policy lookups based several aspects of the route lookup process:

- Limiting the number of stages per message lookup—The Net-Net SBC can limit to the number of additional local policy lookup stages it will perform received message to a maximum of 5. This is configured with the **additional lp lookups**

parameter. Leaving this parameter at its default value of 0 essentially disables multistaged local policy lookups.

- Limiting the number of routes per Local Policy lookup—The Net-Net SBC can limit the number of route results to use as returned for each Local-Policy lookup. This is configured with the **max lp lookups routes per lookup** parameter. Leaving this parameter at its default value of 0 places no limit on the number of returned routes the Net-Net SBC can try.
- Limiting the total number of routes for all local policy lookups per message request—The Net-Net SBC can limit the number of route returned in total across all lookups for a given request, including additional stages. This is configured with the **total lp routes** parameter. Leaving this parameter at its default value of 0 places no limit on the number of returned routes the Net-Net SBC can try. This parameter overrides any configured options.

Additionally, the Net-Net SBC monitors for local policy lookup loops which could cause a significant deterioration in performance. If a loop is found, the Net-Net SBC stops trying the looping route list and proceeds to try any remaining routes..

ACLI Configuration and Examples

To set up your local policy attributes for routing using the TO header:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **local-policy** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit a local policy.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. Type **policy-attributes** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.


```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```
5. **next-hop**—This is the next signaling host and/or object to query. This parameter can be configured as an IP address, ENUM server, or LRT. You can also add a lookup key to an ENUM server or LRT lookup with the following syntax:


```
next-hop enum:ENUM-object;key=$TO
```
6. **terminate-recursion**—Set this parameter to **enabled** to terminate local policy route recursion when the current stage completes.
7. **lookup**—Leave this parameter at the default **single** for single stage local policy routing or set it to **multi** to enable multistage local policy routing.
8. **next-key**—Set this parameter to \$TO, \$FROM, or \$PAI if you wish to override the recently-returned lookup key value for the next stage.
9. Save and activate your configuration.

Maintenance and Troubleshooting

The **show sip policy** command includes four additional counters that refer to single and multistage local policy lookups. All counters are reported for the recent period, and lifetime total and lifetime period maximum. These counters are:

- Local Policy Inits—Number of times the Net-Net SBC makes an initial local policy lookup.
- Local Policy Results Max—Number of times the Net-Net SBC truncated the number of routes returned for a local policy lookup because the maximum number of routes per local policy lookup (**max lp lookups routes per lookup**) threshold was reached.
- Local Policy Exceeded—Number of times the Net-Net SBC truncated the number of routes returned for a local policy lookup because the maximum number of routes per message request (**total lp routes**) threshold was reached.
- Local Policy Loops—Number of times the Net-Net SBC detected a loop while performing a multistage local policy lookup.

Management: SNMP

Traps

An SNMP trap is generated to notify that the limit on the **additional lp lookups** threshold has been reached during the recent window period. This trap occurs a maximum of once during a window period.

```
apSysMgmtLPLookupExceededTrap NOTIFICATION-TYPE
```

```
STATUS    current
```

```
DESCRIPTION
```

```
" The trap will be generated the first time the additional Local
Policy Lookups limit is reached is in the recent window period. The
trap will only occur once during a window period."
```

```
::= { apSystemManagementMonitors 65}
```

Routing-based RN and CIC

When the Net-Net SBC performs local policy routing, it selects local policy entries based on from addresses, to addresses, and source realms. All three are configurable in the local policy configuration. The to addresses can either be the username in a Request-URI (if it is an E.164/phone number format), or the request-URI's hostname or IP address. The Net-Net SBC sorts matching local policies based on policy attribute entries. A policy attribute defines a next hop, which can be a session agent or a session agent group. Alternatively, the next hop might define an ENUM server group or local route table to use to find the next hop.

If the routing-based RN and CIC feature is not enabled, the Net-Net SBC performs the subsequent ENUM query or local route table lookup using the Request-URI's username, if it is a telephone number (TN). The TN is the normalized user part of the Request-URI, ignoring any user parameters or non-digit characters.

If the routing-based RN and CIC feature is enabled, the Net-Net SBC instead performs the ENUM or local route table lookup based on a user parameter, which is useful for lookups based on routing number (RN) or carrier identification code (CIC):

- An RN is a number that identifies terminating switch nodes in Number Portability scenarios when the original TN has been moved to the switch defined by the RN.

- A CIC is the globally unique number of the terminating carrier to which a ported number has been moved.

In applications where the Net-Net SBC is given the RN or the CIC in the Request-URI, this feature is useful because the Net-Net SBC can perform an additional ENUM or local route table lookup to find the next hop to the RN or the CIC. Typically, ENUM servers have imported Number Portability data with which to respond to the Net-Net SBC query, and (for example) the Net-Net SBC can use local route tables for storing CIC values for direct carrier hand-off.

Even with this feature enabled, the Net-Net SBC still performs local policy match selection based on the TN. This feature only uses the RN or CIC user-parameter for the ENUM or local route table lookup after the local policy and policy attributes have been selected.

ACLI Instructions and Examples

This section shows you how to specify that a set of local policy attributes should use an RN for lookup. You can also set this value to CIC, or to any value you require.

Setting the Lookup Key

You can set the lookup key to an RN in the local policy attributes' **next-hop** parameter.

To set the lookup key to RN:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **local-policy** and press <Enter>.


```
ACMEPACKET(session-router)# local-policy
ACMEPACKET(local-policy)#
```
4. Type **policy-attributes** and press <Enter>.


```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#
```
5. **next-hop**—In the **next-hop** parameter—after the kind of ENUM service used—type a colon (:). Then, without spaces, type in **key=rn** and press <Enter>.


```
ACMEPACKET(local-policy-attributes)# next-hop lrt:lookup;key=rn
```
6. Save and activate your configuration.

Introduction

This chapter describes the Net-Net 9200's number translations feature.

About Number Translation

Net-Net 9200 number translation is used to change a Layer5 endpoint name according to prescribed rules. Number translations can be performed on both the inbound and the outbound call legs independently, before and after routing occurs.

Number translation takes place twice for SIP calls. The first number translation is applied to the incoming leg of the call, before the outgoing route is selected. The second number translation is applied to the outgoing leg of the call after the outgoing route is selected.

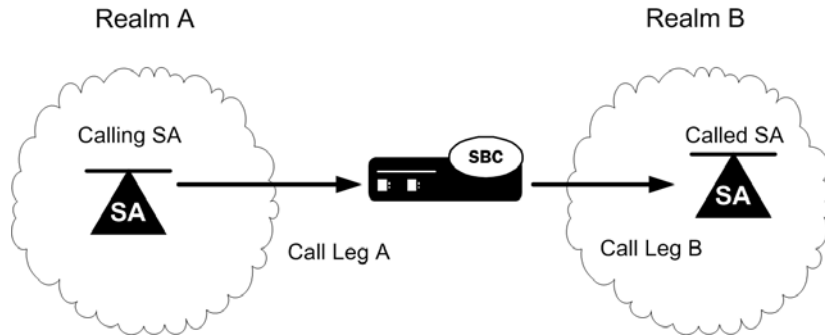
Number translation can be used to strip address prefixes added by external gateways. It can also be used to add a string tag to an address in order to implement a local policy routing scheme, and then remove the tag upon egress from the Net-Net 9200. The most common use of number translation is to add or remove a "1" or a "+" from a phone number sent from or addressed to a device.

Number Translation Implementation

Net-Net 9200 number translations are implemented in three steps. First, the individual number translation rules are defined in the translation rules subelement. Next, the established rules are grouped in a specified order to apply to calling and called numbers. This second step occurs in the session translation element. Finally, session translations are attached to either session agents or realms in the session agent element or realm configuration element.

Number translations attached to session agents take precedence over number translations attached to realms. If no number translation is applied to a session agent, then the Net-Net 9200 will use the number translation applied to a realm. If a number translation is applied to both a realm and session agent, the translation attached to the session agent will apply. If session agents and realms have no associated translations, then all numbers will remain in their original forms as they pass through the Net-Net 9200.

Within each realm or session agent, the number translation is applied to either the incoming or outgoing call leg. This distinction between incoming and outgoing calls is made from the point of view of the Net-Net 9200. The following diagram illustrates the number translation concept.



Leg	Calling SA	Called SA	Realm A	Realm B
A	IN Translation ID		IN Translation ID	
B	OUT Translation ID		OUT Translation ID	

The table above shows you which parameters to apply a session translation ID in order to affect the corresponding leg of the call as shown in the illustration.

Number Translation in SIP URIs

Number translations only change the user portion of the URI. A typical SIP URI looks like `sip:user@hostname`. The user portion can take the form of either a phone number or any other string used for identification purposes.

Within the SIP header exists a Request URI, a To URI, and a From URI. The session translation element's rules calling parameter modifies the From URI, while the rules called parameter modifies the Request URI and the To URI.

Number Translation Configuration Overview

This section describes the procedure to create and apply number translations on the Net-Net 9200.

Configuring the number translation feature requires the following steps:

1. Configure individual translation rules in the translation rules element.
2. Group these rules for use in the session translation element.
3. Apply these groups of rules on a per session agent or per realm basis using the appropriate fields in the session agent or realm configuration elements.

Translation Rules

The translation rules subelement is where the actual translation rules are created. The fields within this element specify the type of translation to be performed, the addition or deletion to be made, and where in the address that change takes place. Translations are not applied to any realm or session agent in this element.

When creating translation rules, first determine the type of translation to perform. The following table lists and describes the three types of number translations.

Value	Description
add	This translation type adds a character or string of characters to the address.
delete	This translation type deletes a character or string of characters from the address.
replace	This translation type replaces a character or string of characters within the address. Replace works by first applying the delete parameter then by applying the add parameter.

After you set the translation type, you define the string to add or delete. The wildcard term for a string to delete is the at-sign, "@". Finally, you specify the character position in the address to make the addition or deletion.

The character position where an add or delete occurs is called an index. The index starts at 0 (immediately before the leftmost character) and increases by 1 for every position to the right you move. In order to specify the final position in an address, use the dollar-sign, "\$".

To create a translation rule that deletes a string:

1. Enter a descriptive name for this translation in the ID field.
2. If you are deleting a specific string, enter it in the delete string field.
If you are deleting a portion of the address string, enter the index number in the delete index field. For this type of deletion, remember to enter the number of characters you are deleting in the form of at-signs "@" in the delete string field.
The first matched string will be deleted, any remaining strings that match will remain. For example, if the address is 187521865 and the string to delete is "18," only the first instance of "18" will be deleted. The second instance will remain after translation.

To create a translation rule that adds a string:

1. Enter a descriptive name for this translation in the ID field.
2. Enter the string you want to add in the add string field.
3. Enter the index of the string insertion in the add-index parameter. If you want to add a string at the end of an number, enter a dollar-sign "\$" in the add index field.

To create a translation rule that replaces a string:

A string replacement involves deleting a string followed by adding a string in the removed string's place. The index is not used when replacing a string.

1. Enter a descriptive name for this translation in the ID field.
2. Enter the string you want to delete in the delete string field.
3. Enter the string you want to add in the add string field.

Session Translation

A session translation defines how translation rules are applied to calling and called numbers. Multiple translation rules can be referenced and applied using this

element, which groups rules together and allows them to be referenced by one identifier.

There are two parameters in the session translation element. The rules calling parameter lists the translation rules to be applied to the calling number. The rules called parameter lists of translation rules to be applied to the called number.

The Net-Net 9200 applies the translation rules in the order in which they are entered. They are applied cumulatively. For example, if this field is configured with a value of “rule1 rule2 rule3”, rule1 will be applied to the original number first, rule2 second, and rule3 last.

To configure the session translation element:

1. Enter a descriptive name for this session translation in the ID field.
2. Enter the IDs of existing translation rules in the rules calling parameter. Multiple rules can be entered in this field. The order you enter them in is the order in which they are applied.
3. Enter the IDs of existing translation rules in the rules called parameter. Multiple rules can be entered in this field. The order you enter them in is the order in which they are applied.

Applying Session Translations

Session translations can be applied to both session agents and realms. Both session agents and realms contain the two parameters that denote incoming and outgoing call legs—in translation ID and out translation ID. These two fields are populated with session translation element IDs.

If none of these fields are populated, no number translation will take place and the original address will remain unchanged as it traverses the Net-Net 9200. Further, any session translation applied to a session agent takes precedence over one applied to a realm.

Session Agent

To configure number translation for a session agent:

1. In the session agent element, set the in translation ID and/or the out translation ID to the appropriate ID you configured in the session translation element. There can be only one entry in each of these fields.

Realm**To configure number translation for a realm:**

1. In the realm configuration element, set the in translation ID and/or the out translation ID to the appropriate ID you configured in the session translation element. There can be only one entry in each of these fields.

Configuring Number Translation

This section explains how to configure number translation. It also provides sample configurations for your reference.

ACLI Instructions and Examples

This section describes how to configure translation rules.

To create a translation rule:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session router configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **translation-rules** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# translation-rules
ACMEPACKET(translation-rules)#
```

From this point, you can configure translation rules parameters. To view all translation rules parameters, enter a **?** at the system prompt. The following is an example what a translation rule configuration might look like. Parameters not described in this section are omitted below.

```
translation-rules
  id          addplus1
  type        add
  add-string  +1
  add-index   0
  delete-string
  delete-index 0
```

Translation Rules**Set the following parameters to configure a translation rule:**

1. **ID**—Set a descriptive ID name for this translation rule
2. **type**—Set this parameter to **add**, **delete**, or **replace**.
3. **description**—Enter a text description of the rule.
4. **add-string**—Enter the string you want to add to the address.
5. **add-index**—Enter the position, 0 being the left most position, where you want to add the string defined in the **add-string** parameter.
6. **delete-string**—Enter the string you want to delete from the address. The at-sign "@" is used as a wildcard.

7. **delete-index**—Enter the position, 0 being the left most spot, where you want to delete the string defined in the **delete-string** parameter. This parameter is only used if the **delete-string** parameter is set to one or more at-signs.

Session Translation

To configure session translations:

1. Exit out of the translation rules element and enter the session translation element.

```
ACMEPACKET(translation-rules)# exit
ACMEPACKET(session-router)# session-translation
ACMEPACKET(session-translation)#
```

From this point, you can configure the session translation element. To view all session translation parameters, enter a ? at the system prompt. The following is an example what a session translation configuration might look like:

```
session-translation
    id                               lrules-out
    rules-calling                    rule1 rule2 rule3
    rules-called                      addplus1
```

1. **ID**—Set a descriptive ID name for this session translation.
2. **rules-calling**—Enter the rules calling in the order in which they should be applied. Multiple rules should be included in quotes and separated by spaces.
ACMEPACKET(session-translation)# rules-calling "rule1 rule2 rule3"
3. **rules-called**—Enter the rules called in the order in which they should be applied. Multiple rules should be included in quotes and separated by spaces.

Number Translation Application

To complete your number translation configuration, you must enter into a **realm-config** or **session-agent** element and assign **session-translations** there.

To move from the session-translation element to the session-agent element:

1. Exit out of the session translation element and enter the session agent element.

```
ACMEPACKET(session-translation)# exit
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

OR

To move from the session-translation element to the realm-config element:

1. Exit from the session translation element to the configuration path.

```
ACMEPACKET(session-translation)# exit
ACMEPACKET(session-router)# exit
2. Navigate to the realm-config element located in the media-manager path.
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```


In both **realm-config** or **session agent** elements, you must specify an **in-translationid** and/or an **out-translationid** in order to configure the number translation.

```

session-agent
  in-translationid
  out-translationid          lrules-out

realm-config
  in-translationid
  out-translationid          lrules-out

```

Set the following parameters to configure a translation rule:

1. **in-translationid**—Enter the configured session translation that you want to affect the incoming traffic in this parameter.
2. **out-translationid**—Enter the configured session translation that you want to affect the outgoing traffic in this parameter.

Other Translations

SIP NAT Translations

There are other translations that occur by way of SIP NAT functionality acting on the SIP R-URI, From-URI, and To URI headers. The translation of URIs in the SIP message occurs as messages are received and sent from the Net-Net system's SIP proxy. These translations create a bridge between the external and home realms and remove all references to the original IPv4 addressing from the packets sent to the destination network.

The purpose of this translation is to prevent private IPv4 addresses from appearing in SIP message URIs while traveling through the public network. This aspect of the SIP NAT's functionality involves either translating the private address to a public address or encrypting the private address into the URI.

For information about configuring these additional number mappings, see the *SIP Signaling Service* chapter of this guide.

FQDN Mapping

The Net-Net 9200 maps FQDNs that appear in certain headers of incoming SIP messages to the IPv4 address that the Net-Net 9200 inserts in outgoing SIP contact headers. The mapped FQDNs are restored in the SIP headers in messages that are sent back to the originator.

This feature is useful to carriers that use IPv4 addresses in the SIP From address to create trunk groups in a PSX for routing purposes. When the carrier's peer uses FQDNs, the carrier is forced to create trunk groups for each possible FQDN that it might receive from a given peer. Similarly, this can apply to SIP Contact and P-asserted-identity headers.

For information about configuring these additional number mappings, see the *SIP Signaling Service* chapter of this guide.

13 Admission Control and QoS Reporting

Overview

This chapter describes how to configure the Net-Net 9200 for call admission control and Quality of Service (QoS) monitoring. Call admission control lets you manage call traffic based on several different policies. It is aimed at managing call admission rates in the network, enabling you to maintain suitable QoS levels. A new call is admitted only if it meets the requirements.

QoS reporting provides you with real-time evaluation of network and route performance. It lets you contrast internal domain and external domain performance and facilitates SLA verification and traffic engineering.

About Call Admission Control

The Net-Net 9200 provides call admission control capabilities based on the following policies:

- Bandwidth (single and multi-level policies)
- Session capacity
- Session rate (sustained and burst)

Note: In order to provide admission control for networks to which the Net-Net system is not directly connected, you need to define multiple realms per network interface.

Bandwidth-Based Admission Control

The Net-Net 9200 is a policy enforcement point for bandwidth-based call admission control. Sessions are admitted or rejected based on bandwidth policies, which are configured on the Net-Net 9200 for each realm.

To manage bandwidth consumption of a network's overall capacity, you can configure aggregate bandwidth policies for each realm. See *Configuring Realms* for additional information.

As the Net-Net 9200 processes call requests to and from a particular realm, the bandwidth consumed for the call is decremented from the bandwidth pool for that realm. The Net-Net 9200 determines the required bandwidth from the SDP information for SIP. Any request that would cause the bandwidth constraint to be exceeded is rejected with a SIP 503 Service Unavailable.

For example, if an incoming SIP message requests PCMU for a payload/encoding name, a zero (0) payload type, and an 8000 cycle clock rate, the Net-Net 9200 must determine how much bandwidth is needed.

To accomplish this task, the system checks the media profile values and reserves the bandwidth required for flows. If the required bandwidth for the new flow exceeds the available bandwidth at the time of the request, the Net-Net system rejects the session.

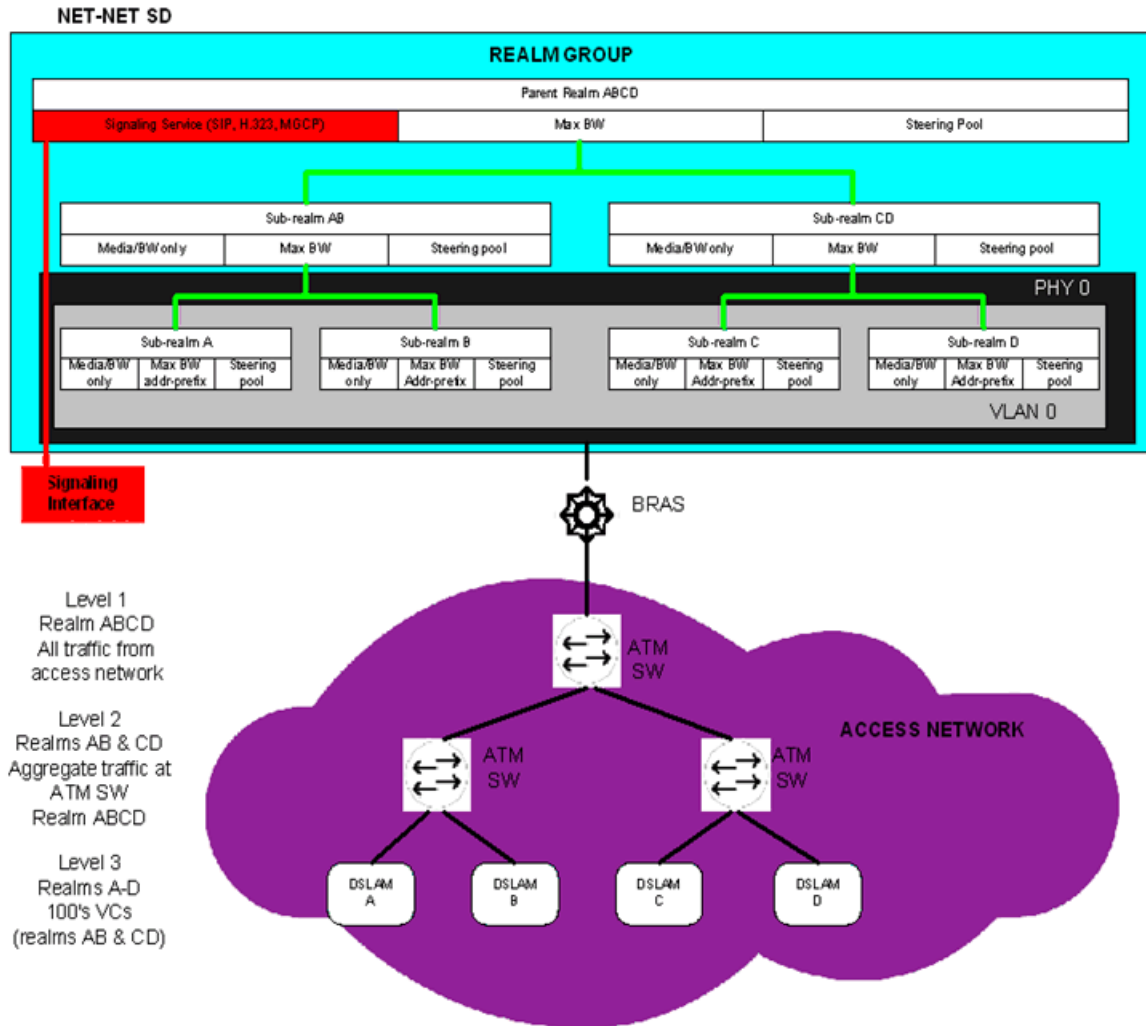
With these mechanisms, the Net-Net system provides bandwidth-based admission control.

Multi-Level Bandwidth Policy Nesting

Multi-level nesting of bandwidth policy enforcement addresses the following issues:

- Bandwidth over-subscription: access or transit transport networks are aggregated and/or oversubscribed. For example, digital subscriber lines (DSL), Frame Relay (FR), and Asynchronous Transfer Mode (ATM). Admission control policies must reflect access network topology.
- Bandwidth partitioning for multiple services: access or transit bandwidth is partitioned among multiple service profiles in the same customer network.
- Multi-site VPN environments: admission control must be applied at the site level as well as the VPN level.

The following example illustrates different scenarios; in each there are two or more levels of admission control required. Nested admission control is best depicted by the DSL broadband example.



In DSL access networks, ATM network bandwidth is typically oversubscribed at rates up to 400/1. At Level 3 (above), hundreds of users virtual circuits (VCs) are aggregated to a smaller set of virtual paths (VPs) at each DSLAM. At Level 2, many virtual paths are aggregated at the first ATM switch. Finally, at Level 1, all traffic from all subscribers in the access network is aggregated at the BRAS. Each level of aggregation is oversubscribed, creating the need to perform admission control at each level.

From a Net-Net 9200 perspective, multiple tiers of realms are supported, each with its unique bandwidth policy. Only the lowest order realm (Level 3) requires an address prefix (that assigned to the DSLAM) that must be used by the Net-Net 9200 to determine in which realm a user resides. When a call request to or from a particular user is received, the Net-Net 9200 checks each realm in the path to determine whether sufficient bandwidth is available to place the call.

Session Capacity- and Rate-based Admission Control

A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. You can define concurrent session capacity and rate attributes for each session agent.

You can configure a set of attributes and constraints for each session agent to support session access control. In this configuration, the Net-Net 9200 only accepts requests from configured session agents. And you can set up session admission control so that the Net-Net 9200 limits the number of concurrent inbound and outbound sessions for any known service element.

The Net-Net 9200 denies a call request to any destination that has exceeded its configured policies for session capacity and session rate. The Net-Net 9200 might reject the call request back to the originator. If multiple destinations are available, the Net-Net 9200 will check current capacity and rate for each destination and attempt to route the call only to destinations whose policy limits have not been reached.

Configuring Admission Control

This section explains how to session agents for admission control.

ACLI Instructions and Examples

To use admission control based on session rate, you need to configure session agent session rate constraints.

To configure session rates:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. Enable session agent constraints and then configure the parameters related to session capacity or session rate to set admission control.
 - 4a. **constraints**—Ensure this parameter is set to enabled. From here you can either configure admission control based on session capacity, session rates, or both.

Session Capacity

You configure the session agent constraints that apply to session capacity.

To configure session capacity:

1. Ensure you have enabled session agent constraints first.
2. **max-sessions**—Set the maximum number of sessions (inbound and outbound) allowed by the session agent. The range of values is:
 - minimum: 0 is the default value
 - maximum: 999999999
3. **max-outbound-sessions**—Enter the maximum number of concurrent outbound sessions (outbound from the Net-Net 9200) that are allowed from this session agent. The range of values is:
 - minimum: zero (0) is the default value

- maximum: 999999999

Note: The number you enter here cannot be larger than the number you entered for max-sessions.

Session Rates

You configure the session agent constraints that apply to session rates. Ensure you have enabled session agent constraints first.

To configure session rates:

1. **max-burst-rate**—Enter a number to set how many SIP session invitations this session agent can send or receive (per second) within the configured burst rate window value. The range of values is:
 - minimum: 0 session invitations per second (default)
 - maximum: 999999999 session invitations per second
2. **max-sustain-rate**—Enter a number to set the maximum rate of session invitations (per second) this session agent can send or receive within the current window. The range of values is:
 - minimum: 0 session invitations per second (default)
 - maximum: 999999999 session invitations per second

Note: The number you enter here must be larger than the number you enter for max-burst-rate.

For the sustained rate, the Net-Net 9200 maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

3. **burst-rate-window**—Enter a number to set the burst window period (in seconds) that is used to measure the burst rate. The term window refers to the period of time over which the burst rate is computed. (Refer to max-burst-rate information.)

The range of values is:

- minimum: 0 seconds (default)
- maximum: 999999999 seconds

4. **sustain-rate-window**—Enter a number to set the sustained window period (in seconds) that is used to measure the sustained rate. (Refer to the max-sustain-rate information). The default value is zero (0), which disables the functionality.

The range of values is:

- minimum: 10
- maximum: 999999999 seconds

Note: If you are going to use this parameter, you must set it to a minimum value of 10.

Note: The value you set here must be higher than or equal to the value you set for the burst rate window.

The following example shows session agent constraints that are enabled and the session capacity parameters have been configured. All other session agent parameters have been omitted for brevity.

```

session-agent
  constraints          enabled
  max-sessions         355
  max-outbound-sessions 355

```

The following example shows session agent constraints are enabled and the session rate parameters have been configured. All other session agent parameters have been omitted for brevity.

```

session-agent
  constraints          enabled
  max-burst-rate       50
  max-sustain-rate     3000
  burst-rate-window    60
  sustain-rate-window  3600

```

Configuring Realm Bandwidth

To configure admission control based on bandwidth, you set the maximum bandwidth parameters in the realm configuration.

To configure realm bandwidth:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Configure the maximum bandwidth.
 - 4a. **max-bandwidth**—Enter a number that sets the maximum bandwidth for dynamic flows to/from the realm in kilobits (Kbps) per second.

Available values include:

 - minimum: 0 Kbps
 - maximum: 999999999 Kbps

The following example shows the maximum bandwidth for the realm has been configured. All other realm parameters have been omitted for brevity.

```

realm-config
  max-bandwidth      64000

```

SIP Admission Control

If a request does not meet the criteria of the option you choose here, it is rejected with a 403 (Forbidden) response.

To configure admission control:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


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


```
ACMEPACKET(session-router)# sip-interface
```

```
ACMEPACKET(sip-interface)#
```

4. Type **sip-ports** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(sip-interface)# sip-port
```

```
ACMEPACKET(sip-port)#
```

5. Set the criteria for admission control.

- 5a. **allow-anonymous**—Enter the anonymous connection option (mode) you want applied when SIP requests are processed.

The following table lists the available options.:

Option	Description
all	No ACL is applied and all anonymous connections are allowed.
agents only	Only requests from configured session agents are processed. The Net-Net 9200 responds to all other requests with a <i>forbidden</i> response.
realm prefix	Only requests from session agents and addresses matching the realm's address prefix are processed. All other requests are rejected with a 403 (Forbidden) response.
registered	Only requests from session agents and registered endpoints are processed. REGISTER allowed from any endpoint.
registered prefix	Only requests from session agent and registered endpoint addresses that match the realm's realm prefix are processed.

The following example shows the **allow-anonymous** parameter that has been configured to allow only requests from session agents and registered endpoints. All other SIP interface>SIP port parameters following the **allow-anonymous** parameters are omitted for brevity.

```

sip-port
    address
    port 5060
    transport-protocol UDP
    allow-anonymous registered

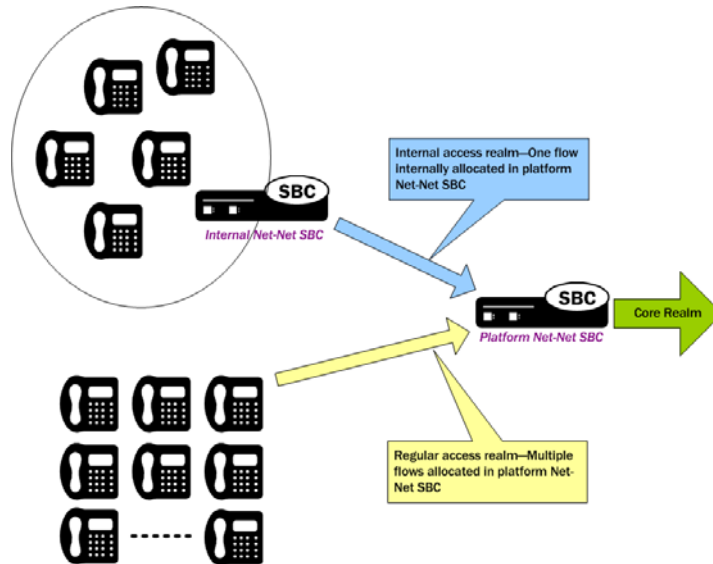
```

Session Agent Minimum Reserved Bandwidth

You can assign session agents minimum bandwidth, applicable in access Net-Net SBC deployments. Assigning a session agent minimum bandwidth can prevent overloading other network devices—such as another Net-Net SBC configured as a session agent. Doing so assures signaling bandwidth and availability to the endpoints behind this Net-Net SBC.

In the following diagram, the internal Net-Net SBC is configured as a session agent on the platform Net-Net SBC (which conveys traffic to the core realm). Setting up bandwidth reservation allows for the creation of only one allocated flow, and secures bandwidth for all the SIP clients behind the internal Net-Net SBC. Contrast this

scenario with the one where the platform Net-Net SBC must allocate multiple flows for many SIP clients.



How It Works

When you configure minimum reserved bandwidth for session agent to a non-zero value, the Net-Net SBC allocates a separate pipe for per session agent. This is achieved by setting up an access control configuration in a specific way, instructing the Net-Net SBC to use a minimum number of transmission timeslots the individual pipe is guaranteed to receive.

This feature works across all signaling services: SIP or H.323. No more than 4000 session pipes are supported.

ACLI Instructions and Examples

For the feature to work, you must set up an access control configuration with the settings required in the instructions and examples below.

To configure minimum reserved bandwidth for session agents:

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

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

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **access-control** and press <Enter>.

```
ACMEPACKET(session-router)# access-control
ACMEPACKET(access-control)#
```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **realm-id**—Enter the name of a valid realm.
5. **application-protocol**—Enter a valid application protocol. There is no default for this parameter, and valid values are: **SIP** or **H.323**.
6. **access**—Set this parameter to **permit** (default).

7. **trust-level**—Set this parameter to **high**, changing it from the default (**none**).
8. **minimum-reserved-bandwidth**—Enter the minimum reserved bandwidth you want for the session agent, and that will trigger the creation of a separate pipe for it. Only a non-zero value will allow the feature to work properly, along with the other required values set out in these instructions. The default is 0, and the maximum is 999999999.
9. Save and activate your configuration.

Aggregate Session Constraints for SIP

You can configure a full suite of session constraints and then apply them to a SIP interface. The session constraints configuration contains many of the same parameters as the session agent, so you can configure a group of constraints and then apply them to a SIP interface.

Session constraints configured natively on a SIP interface take precedence over session constraints applied in aggregate to a SIP interface.

The SIP interface configuration's **constraint-name** parameter invokes the session constraint configuration you want to apply. Using the constraints you have set up, the Net-Net SBC checks and limits traffic according to those settings for the SIP interface. Of course, if you do not set up the session constraints or you do not apply them in the SIP interface, then that SIP interface will be unconstrained. If you apply a single session-constraint element to multiple SIP interfaces, each SIP interface will maintain its own copy of the session-constraint.

SIP interfaces now have two states: "In Service" and "Constraints Exceeded." When any one of the constraints is exceeded, the status of the SIP interface changes to "Constraints Exceeded" and remains in that state until the time-to-resume period ends. The session constraint timers that apply to the SIP interface are the time-to-resume, burst window, and sustain window.

ACLI Instructions and Examples

This section shows you how to configure aggregate session constraints and then apply them to a SIP interface.

Configuring Session Constraints

The session constraints configuration contains many of the same parameters as the session agent does; it also incorporates the changes to the session agent parameters that are described in this section.

To configure the session constraints configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-constraints** and press <Enter>.


```
ACMEPACKET(session-router)# session-constraints
```
4. **name**—Enter the name for this session constraints configuration; this is a unique identifier that you will use in the SIP interface when you want the session constraints applied there. This is a required parameter that has no default.

5. **state**—To use these session constraints, leave this parameter set to **enabled** (default). To stop using them, change the value to **disabled**.
6. **max-sessions**—Enter the maximum sessions allowed for this constraint. The minimum value is 0 (default), and the maximum is 999999999.
7. **max-outbound-sessions**—Enter the maximum outbound sessions allowed for this constraint. The minimum value is 0 (default), and the maximum is 999999999.
8. **max-inbound-sessions**—Enter the maximum inbound sessions allowed for this constraint. The minimum value is 0 (default), and the maximum is 999999999.
9. **max-burst-rate**—Enter the maximum burst rate (invites per second) allowed for this constraint. This value should be the sum of the **max-inbound-burst-rate** and the **max-outbound-burst-rate**. The minimum value is 0 (default), and the maximum is 999999999.
10. **max-sustain-rate**—Enter the maximum rate of session invitations per second allowed within the current window for this constraint. The minimum value is 0 (default), and the maximum is 999999999.

For the sustained rate, the Net-Net SBC maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

11. **max-inbound-burst-rate**—Enter the maximum inbound burst rate (number of session invitations per second) for this constraint. The minimum value for this parameter is 0 (default), and the maximum is 999999999.
12. **max-inbound-sustain-rate**—Enter the maximum inbound sustain rate (of session invitations allowed within the current window) for this constraint. The minimum value for this parameter is 0 (default), and the maximum is 999999999.

For the sustained rate, the Net-Net SBC maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

13. **max-outbound-burst-rate**—Enter the maximum outbound burst rate (number of session invitations per second) for this constraint. The minimum value for this parameter is 0 (default), and the maximum is 999999999.
14. **max-outbound-sustain-rate**—Enter the maximum outbound sustain rate (of session invitations allowed within the current window) for this constraint. The minimum value for this parameter is 0 (default), and the maximum is 999999999.

For the sustained rate, the Net-Net SBC maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes.

15. **min-seizures**—Enter the minimum number of seizures for a no-answer scenario. The default is 5 and the valid range is 0-999999.
16. **min-asr**—Enter the minimum ASR in percentage. The minimum value for this parameter is 0 (default), and the maximum is 999999.
17. **time-to-resume**—Enter the number of seconds after which the SA (Session Agent) is put back in service (after the SA is taken OOS (Out Of Service) because it exceeded some constraint). The minimum value for this parameter is 0 (default), and the maximum is 999999999.
18. **ttr-no-response**—Enter the time delay in seconds to wait before the SA (Session Agent) is put back in service (after the SA is taken OOS (Out Of Service) because it did not respond to the Net-Net SBC). The default value is zero (0). The valid range is:

- Minimum—0
 - Maximum—999999999
19. **burst-rate-window**—Enter the time in seconds that you want to use to measure the burst rate; the “window” is the time over which the burst rate is calculated, and is used for the over all burst rate as well as the inbound and outbound burst rates. The minimum value for this parameter is 0 (default), and the maximum is 999999999.
 20. **sustain-rate-window**—Enter the time in seconds used to measure the sustained rate; the “window” is the time over which the sustained rate is calculated, and is used for the over all sustained rate as well as the inbound and outbound sustained rates. The minimum value for this parameter is 0 (default), and the maximum is 999999999.
 21. Save your work using the ACLI **done** command.
 22. Save and activate your configuration.

Applying Session Constraints in a SIP Interfaces

In the SIP interface, there is a new parameter that allows you to use a set of session constraints for that interface; the parameter is called constraint-name.

To apply session constraints to a SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
```
4. **session-constraints**—Enter the name of the session constraints configuration that you want to apply to this SIP interface. There is no default for this parameter.
5. Save your work using the ACLI **done** command.
6. Save and activate your configuration.

SIP Per-User CAC

The Net-Net SBC’s call admission control (CAC) supports an enhanced degree of granularity for SIP sessions.

Without this feature enabled, the Net-Net SBC performs call admission control (CAC) based on:

- Bandwidth limits configured in realms and nested realms
- Number of media flows available through the steering pool per realm
- Number of inbound sessions configured for a SIP session agent
- Number of total sessions (inbound and outbound) per SIP session agent
- Use of the Net-Net SBC’s support for common open policy service (COPS), allowing the Net-Net SBC to perform CAC based on the policies hosted in an external policy server

These methods provide a basic level of call admission control in order to ensure that a SIP session agent’s capacity is not exceeded. You can also ensure that signaling and media bandwidth capacities are not exceeded for physical trunks and peers.

With this feature enabled, the Net-Net SBC changes behavior so that it will only allow the configured number of calls or total bandwidth to and from each user in a particular realm. The overall realm bandwidth and steering pool limits still apply, and as before, the Net-Net SBC still rejects users who might be within their CAC limitations if accepting them would exceed the bandwidth limitations for parent or child realms and steering pools.

For SIP sessions, the Net-Net SBC keeps track of the amount of bandwidth a user consumes and the number of active sessions per address of record (AoR) or per IP address, depending on the CAC mode you select (either `aor` or `ip`). When an endpoint registers with the Net-Net SBC, the Net-Net SBC allots it a total amount of bandwidth and total number of sessions.

How It Works

This section describes the details of how SIP per user CAC works.

You should note that the functionality this section describes only works if you enable registration caching on your Net-Net SBC.

For SIP sessions, the Net-Net SBC now keeps track of the amount of bandwidth a user consumes and the number of active sessions per address of record (AoR) or per IP address, depending on the CAC mode you select (either `aor` or `ip`). When an endpoint registers with the Net-Net SBC, the Net-Net SBC allots it a total amount of bandwidth and total number of sessions.

Per User CAC Modes

There are three modes that you can set for this feature, and each has an impact on how the other two per-user-CAC parameters are implemented:

- `none`—No per user CAC is performed for users in the realm.
- `aor`—The Net-Net SBC performs per user CAC according to the AoR and the contact associated with that AoR for users in the realm.
- `ip`—The Net-Net SBC performs per user CAC according to the IP address and all endpoints that are sending REGISTER messages from the IP address for users in the realm.

Per User CAC Sessions

You can set the number of CAC for sessions per user in the realm configuration. Depending on the CAC mode you set, the sessions are shared between contacts for the same AoR or the endpoints behind the same IP address.

When it receives an INVITE, the Net-Net SBC determines the registration entry for the calling endpoint and the registration for the called endpoint. It then decides if a session can be established between the two. If it can, the Net-Net SBC establishes the session and changes the active session count for the calling and called endpoints. The count is returned to its original value once the session is terminated.

Per User CAC Bandwidth

You can set the per user CAC bandwidth in realm configuration, too, and it is handled much the same way that the sessions are handled. That is, depending on the CAC mode you set, the bandwidth is shared between contacts for the AoR or the endpoints behind the same IP address. All endpoints must be registered with the Net-Net SBC.

When it receives a Request with SDP, the Net-Net SBC checks to see if there is enough bandwidth for the calling endpoint and for the called endpoint. The Net-Net SBC assumes that the bandwidth usage is symmetric, and it uses the maximum

bandwidth configured for the codec that it finds in the Request. In the event that there are multiple streams, the Net-Net SBC determines the total bandwidth required for all of the streams. If the required bandwidth exceeds what is available for either endpoint, the Net-Net SBC rejects the call (with a 503 error response). If the amount of available bandwidth is sufficient, then the used bandwidth value is increased for both the registered endpoints: calling and called. Any mid-session requests for changes in bandwidth, such as those caused by modifications in codec use, are handled the same way.

The Net-Net SBC also keeps track of the bandwidth usage on a global level. When the call terminates, the bandwidth it was consuming is returned to the pool of available bandwidth.

ACLI Instructions and Examples

Note that you must enable registration caching for this feature to work.

To configure SIP per user CAC:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Select the realm where you want to add SIP per user CAC.


```
ACMEPACKET(realm-config)# select
```
5. **user-cac-mode**—Set this parameter to the per user CAC mode that you want to use: **aor** or **ip**. If you do not enter either of these modes, the Net-Net will not perform per user CAC for users in this realm; it will remain set to **none** (default).
6. **user-cac-sessions**—Enter the maximum number of sessions per user for dynamic flows to and from the user. Leaving this parameter set to **0** (default) means that there is unlimited sessions, meaning that the per user CAC feature is disabled in terms of the constraint on sessions. The value range you can enter is a minimum of **0**, and a maximum of **4294967295**.
7. **user-cac-bandwidth**—Enter the maximum bandwidth per user for dynamic flows to and from the user. Leaving this parameter set to **0** (default) means that there is unlimited bandwidth, meaning that the per user CAC feature is disabled in terms of the constraint on bandwidth. The value range you can enter is a minimum of **0**, and a maximum of **4294967295**.
8. Save your work using the ACLI **done** command.
9. Save and activate your configuration.

Monthly Minutes-based CAC

Building on the Net-Net SBC pre-existing call admission control methods, CAC can now be performed based on how many minutes are being used by SIP calls per-realm for a calendar month.

In the realm configuration, you can now set a value representing the maximum number of minutes to use for SIP sessions using that realm. Although the value you

configure is in minutes, the Net-Net SBC performs CAC based on this value to the second.

The Net-Net SBC calculates the number of minutes used when a call completes, and counts both call legs for a call that uses the same realm for ingress and egress. The total time attributed to a call is the amount of time between connection (SIP 200 OK) and disconnect (SIP BYE), regardless of whether media is released or not; there is no pause for calls being placed on hold.

If the number of minutes is exhausted, the Net-Net SBC rejects calls with a SIP 503 Service Unavailable message (including additional information “monthly minutes exceeded”). In the event that the limit is reached mid-call, the Net-Net SBC continues with the call that pushed the realm over its threshold but does not accept new calls. When the limit is exceeded, the Net-Net SBC issues an alarm and sends out a trap including the name of the realm; a trap is also sent when the alarm condition clears.

You can change the value for minutes-based CAC in a realm configuration at any time, though revising the value downward might cause limits to be reached. This value resets to zero (0) at the beginning of every month, and is checkpointed across both system in an HA node. Because this data changes so rapidly, however, the value will not persist across and HA node if both systems undergo simultaneous failure or reboot.

You can use the ACLI **show monthly-minutes <realm-id>** command (where **<realm-id>** is the realm identifier of the specific realm for which you want data) to see how many minutes are configured for a realm, how many of those are still available, and how many calls have been rejected due to exceeding the limit.

ACLI Instructions and Examples

This section shows you how to configure minutes-based CAC for realms and how to display minutes-based CAC data for a specific realm.

Enabling Monthly Minutes-Based CAC

Note that setting the new monthly-minutes parameters to zero (0), or leaving it set to its default of 0, disables this feature.

To configure minutes-based CAC:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Select the realm where you want to add SIP per user CAC.


```
ACMEPACKET(realm-config)# select
```
5. **monthly-minutes**—Enter the number of minutes allowed during a calendar month in this realm for SIP calls. By default, this parameter is set to zero (0), which disabled monthly minutes-based CAC. You can enter a value as high as 4294967295.

6. Save and activate your configuration.

Viewing Monthly Minutes-Based CAC Data

Use the ACLI show monthly-minutes command to see the following information:

- How many minutes are configured for a realm
- How many of those are still available
- How many calls have been rejected due to exceeding the limit

To view information about SIP per user CAC using the IP address mode:

1. In either User or Superuser mode, type **show monthly-minutes <realm-id>**, a <Space>, and the IP address for which you want to view data. Then press <Enter>. The <realm-id> is the realm identifier for the realm identifier of the specific realm for which you want data

ACMEPACKET# **show monthly-minutes private_realm**

Shared CAC for SIP Forked Calls

A forked call is one which has multiple INVITEs for the same call. For example, if an Application Server in the provider core network forks a call attempt, the application server sends several INVITEs for the same call toward the Net-Net SBC. Each INVITE is destined for a unique device that belongs to the same user. Ideally, that user will only answer one device. The Net-Net SBC treats each INVITE as a unique call request.

By default, each of the multiple INVITE forks are checked against CAC bandwidth limits, and thus they each consume bandwidth resources when they are received, even though only one of the forks will succeed in establishing a permanent session. Therefore, for many operators the CAC behavior of the SD is too restrictive and results in rejected call attempts which should have been allowed.

The following diagram shows a forked call scenario. The total bandwidth counted against the realm is 60 kbps. If the realm has a bandwidth ceiling of 50 kbps, one of the INVITEs will be rejected.



You can, however, enable the system to enforce CAC limits only once for SIP forked calls as long as the calls are identified as such, meaning that they will use the same bandwidth resources. The Net-Net SBC counts the forked call's most bandwidth-hungry codec at the time it arrives at the Net-Net SBC. In the above diagram, with shared bandwidth for forked calls enabled, the Net-Net SBC counts 30 kbps against the realm's total bandwidth after that INVITE arrives, even after the first two INVITES have passed into the final realm.

Bandwidth Sharing Scenarios

The following table summarizes how bandwidth would be shared given certain ingress and egress realms with this feature enabled. Realms A and C are call ingress realms.; realms B and D are egress realms. For the bandwidth to be shared, Call A and Call B must have the same forked Call-ID in the P-Multiring-Correlator header and be entering or exiting the Net-Net SBC on the same realm.



ACLI Instructions and Examples

To enable bandwidth sharing of forked calls, set the **forked-cac-bw** parameter in the SIP profile configuration to **shared**. Although there are other parameters available in the SIP profile configuration, you only have to set the **name** and the **forked-cac-bw** values to use this feature. The **forked-cac-bw** parameter can also be configured to inherit so that it will inherit the value from the from realm-config or sip-interface's sip-profile.

After you set up the SIP profile, you apply it to a realm, SIP interface, or session agent.

Configuring a SIP Profile

The SIP profile is an element in the ACLI's **session-router** path, and you can configure multiple SIP profiles.

To configure a SIP profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-profile** and press <Enter>.


```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```
4. **name**—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's **name** when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. **forked-cac-bw**—Set this parameter to **shared** if you want forked sessions to share bandwidth resources, or set it to **per-session** if you want bandwidth to be counted for each session individually. The default value is **inherit** which inherits the value from the from realm-config or sip-interface's sip-profile.
6. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- realm-config
- sip-interface
- session-agent

To apply a SIP profile to a SIP interface:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **sip-profile**—Enter the name of SIP profile configuration that includes the **forked-cac-bandwidth** parameter configured.
5. Save your work.

Conditional Bandwidth CAC for Media Release

The Net-Net SBC supports conditional call admission control (CAC) using the SIP profile configuration. With this feature enabled, you can allow the conditional admission of SIP calls that could potentially have their media released instead of risking the possible rejection of those calls due to internal bandwidth limits.

About Conditional Bandwidth CAC for Media Release

The Net-Net SBC performs bandwidth CAC for SIP per realm, for each Address of Record (AoR) or IP address. The system checks bandwidth limits based on the codecs listed in SDP. If a new SIP INVITE contains codecs in an SDP message that exceed bandwidth available for a given resource, the system rejects that INVITE. This check occurs both on the ingress and egress sides of a call, and both sides must have enough available resources to support the call for it to be admitted.

In the case of calls where media is released, the Net-Net SBC does not count bandwidth consumed by the call. However, this exemption is not given until the media is actually released—and media release conditions are unknown at the time SIP INVITE is admitted. This is because an INVITE received on one side of the Net-Net SBC is only media-released when that INVITE is routed back through the Net-Net SBC as a hairpin or other multi-system media release. So there has to be enough bandwidth for the initial INVITE; otherwise, and even if the INVITE is a candidate for media release, it will be rejected.

When there is a significant volume of such calls—ones that are candidates for media release, but cannot be admitted because of CAC limits—it becomes important to admit them so long as they truly end in media release. This feature thus allows admission of SIP calls that might otherwise be rejected because of bandwidth limitations when the far-end of the call causes media to be released.

Details and Conditions

This feature applies in a two system scenario. In order to track a call as a candidate for provisional media release, the access-side Net-Net SBC adds a Require: header with an option tag to the INVITE or UPDATE message on egress. The option tag is configurable in the sip config option. The default is *com.acmepacket.cac*.

The following sections describe when the SIP INVITE or SIP UPDATE are:

- initially received by the Net-Net SBC
- received by the second Net-Net SBC

INVITEs/UPDATEs Initially Received By Net-Net SBC

When the Net-Net SBC first receives an INVITE or UPDATE message, it considers if it should be admitted provisionally or rejected outright due to CAC bandwidth constraints. If the INVITE or UPDATE is admitted provisionally, a Require: header is inserted on egress from the system.

The Net-Net SBC inserts the Require header on egress under these conditions:

- It receives an INVITE / UPDATE with no or a non-matching Require header.
- The **egress conditional cac admit** parameter in the SIP profile on the egress realm, SIP interface, session agent is set to enabled in the egress realm
- The request would otherwise be rejected because of current bandwidth CAC limits in the ingress OR egress realms
- The call is a candidate for media-release in the ingress realm

A call is considered a candidate for media-release when the ingress realm has any of these parameters set to disabled:

- **mm-in-realm**
- **mm-in-network**
- **mm-same-ip**
- **mm-in-system**

INVITEs/UPDATEs Received by Second Net-Net SBC

The second Net-Net SBC receives the INVITE or UPDATE with the newly inserted Require: header. Standard SIP convention indicates that if the UAS receiving the request does not know how to handle the Require header, the request should be rejected.

When the following three conditions are met, the INVITE is permitted into the system for processing:

- The **ingress conditional cac admit** in the SIP profile on the ingress realm, SIP interface, session agent parameter is set to enabled
- The **con-cac-tag** sip config option is configured to the same value as the received Require header's option tag
- The call is a candidate for media-release

The call is considered a candidate for media-release on the second system (indicated by the **ingress conditional cac admit** parameter is set to enabled) when either the ingress or egress realms have any of these parameters set to disabled:

- **mm-in-realm**
- **mm-in-network**
- **mm-same-ip**
- **mm-in-system**

and the following parameter is set to enabled:

- **msm-release**

If the call, as received by the second system is not considered a candidate for release, the INVITE or UPDATE is failed with a 503 “Insufficient Bandwidth” message.

After the INVITE has been processed by the Net-Net SBC, the Require: header is removed upon egress from the system.

The following diagram shows the two-system scenario:



Conditional Admission with Per-user CAC

In the event that the per-user CAC feature is also being used, and per-user CAC bandwidth is exceeded, the Net-Net SBC also uses this option tag mechanism. However, if the per-user CAC implementation does count bandwidth regardless of media-release, then the Net-Net SBC will reject calls exceeding the per-user CAC limits when it receives them.

On the second system, when the per-user CAC feature is being used, the Net-Net SBC will perform the same option tag mechanism based on if the **ingress conditional cac admit** parameter is enabled.

ACLI Instructions and Examples

You enable this feature by first configuring a SIP profile, and then applying the profile to any of these:

- realm
- SIP interface
- SIP session agent

Configuring a SIP Profile

The SIP profile is an element in the ACLI’s **session-router** path, and you can configure multiple SIP profiles. Though this configuration contains additional

parameters, you do not have to use them for the conditional bandwidth CAC for media release.

To configure a SIP profile:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **account-config** and press <Enter>.


```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```
4. **name**—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's **name** when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. **ingress-conditional-cac-admit**—Set this parameter to enabled to process an INVITE with a Require tag as received on an ingress interface. You can set this parameter to disabled if you do not want to use this feature on the ingress side. You can leave this parameter set to its default of inherit which inherits the value from the from realm-config or sip-interface's sip-profile.
6. **egress-conditional-cac-admit**—Set this parameter to enabled if you want to use conditional bandwidth CAC for media release for calls that are first received by this system. This results in option tags being inserted on the INVITE's egress if the conditional CAC conditions are met. You can set this parameter to disabled if you do not want to use this feature. There is no default for this parameter. You can leave this parameter set to its default of inherit which inherits the value from the from realm-config or sip-interface's sip-profile.
7. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- realm-config
- sip-interface
- session-agent

To apply a SIP profile to a realm:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. **sip-profile**—Enter the name of SIP profile configuration you want to use for conditional bandwidth CAC for media release for this SIP interface. This value is blank by default, but it must be the value of the **name** parameter from a valid SIP profile.
5. Save your work.

Configuring Require Header Option Tag

You may change the Require: header's option tag from the default `com.acmepacket.cac` to one of your own choosing. Remember that both systems' option tags must match exactly.

To configure the Require: header's option tag:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# sip-config
```
4. Use the ACLI **select** command so that you can work with the SIP configuration.


```
ACMEPACKET(sip-config)# select
```
5. **options**—Set the options parameter by typing **+options**, a <Space>, the option name **con-cac-tag=your-new-tag**, and then press <Enter>.


```
ACMEPACKET(sip-config)# options +con-cac-tag=com.test.cac
```
6. Save your work.

CAC, Policing, and Marking for non-Audio/non-Video Media

The Net-Net SBC supports non-AVT (audio-visual transport) media profile and media policy configurations.

In previous releases, the Net-Net SBC only policed media based on average rate limits configured in media profiles, but these are only applied to AVT. And if there are not required bandwidth or average rate limit values set for the media profile, CAC and policing functions are not applied to media—even if the SDP specifies appropriate bandwidth values. Likewise, ToS markings are not applied for non-AVT media, but only for SIP, H.323, and AVT media types.

With this feature addition, you can now enable your Net-Net SBC to handle non-AVT media types like image and text, and use application and data type for policing purposes. Bandwidth CAC support has also been added for non-AVT media types, as has support for the application specific (AS) bandwidth modifier (**b=AS:<value>**) in the SDP with specification of a defined amount of headroom for that value.

Configuring CAC, Policing, and Marking for non-Audio, non-Video Media

In the media profile and the media policy configurations, the following values have been added for the **media-type** parameter:

- application | data | image | text

For the media policy, these new values apply to ToS marking.

Bandwidth CAC Fallback Based on ICMP Failure

For networks where backup links (operating in active-standby mode) from CE-routers to the MPLS backbone are provisioned with less bandwidth than the primary links, the Net-Net SBC can:

- Detect remote link failures
- Trigger bandwidth updates at the realm level when using backup links
- Detect remote link fallback to primary

To do so, the Net-Net SBC monitors the primary link status using ICMP echo requests (or pings). It issues the pings at regular intervals, forming a heartbeat mechanism. The CE-router can respond to these pings on the primary link, which is represented by the WAN IP address. When this link fails over, the backup link assumes the same WAN IP address but is not responsive to the pings. This way, the Net-Net SBC determines failover when the ICMP ping fails.

When there is an ICMP ping failure, the Net-Net SBC adjusts the realm's available bandwidth pool from its maximum bandwidth setting to its fallback setting. If the fallback amount is less than the maximum amount, it is possible for the Net-Net SBC to start rejecting calls. It does so until enough calls are released to free adequate bandwidth to stay under the fallback limit and still accept calls.

ACLI Instructions and Examples

You can set up ICMP heartbeats and fallback bandwidth pools in the realm configuration. Leaving the **icmp-detect-multiplier**, **icmp-advertisement-interval**, or **icmp-target-ip** parameters blank or set to zero turns the feature off.

To enable bandwidth CAC fallback based on ICMP failure:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **icmp-detect-multiplier**—Enter the multiplier you want to use when determining how long to send ICMP pings before considering a target unreachable. This number multiplied by the time you set for the **icmp-advertisement-interval** determines the length of time. For example, if you set this parameter to 10 and the advertisement interval to 20, the Net-Net SBC will send ICMP pings for 120 seconds before declaring the target unreachable.
5. **icmp-advertisement-interval**—Enter the time in seconds between ICMP pings the Net-Net SBC sends to the target. The default is 0.
6. **icmp-target-ip**—Enter the IP address to which the Net-Net SBC should send the ICMP pings so that it can detect when they fail and it needs to switch to the fallback bandwidth for the realm. There is no default.
7. **fallback-bandwidth**—Enter the amount of bandwidth you want available once the Net-Net SBC has determined that the target is unreachable.

If the fallback amount is less than the **max-bandwidth** value, the Net-Net SBC might start to reject calls. It does so until enough calls are released to free adequate bandwidth to stay under the fallback limit and still accept calls.

8. Save and activate your configuration.

Bandwidth CAC for Aggregate Emergency Sessions

You can configure the maximum amount of bandwidth on your Net-Net SBC you want used specifically for priority (emergency) calls in the realm configuration's **max-priority-bandwidth** parameter. You set this limit on a per-realm basis, and the limit is enforced for nested realms. Setting a bandwidth limit specifically for priority calls allows the Net-Net SBC to reject calls exceeding the threshold, and also to accept calls that exceed the bandwidth limit for non-priority calls (set in the **max-bandwidth** parameter).

The bandwidth limit for emergency calls operates in conjunction with the bandwidth limits you can set for all other types of calls. When an emergency call comes in, the Net-Net SBC checks the non-priority bandwidth limit. If bandwidth is sufficient, the call goes through and the Net-Net SBC decrements the bandwidth used from the pool of the amount available.

However, if a priority call exceeds the **max-bandwidth** setting, the Net-Net SBC checks the **max-priority-bandwidth** parameter. If it is within the limit for priority calls, the system allows the call and decrements the amount of used bandwidth from what is available.

When there is not enough bandwidth in either the priority or non-priority pool, the Net-Net SBC rejects the call with the corresponding error code and reason phrase.

Any bandwidth subtracted from either pool during a session is returned to that pool as soon as the session ends.

ACLI Instructions and Example

You configure bandwidth CAC for priority calls on a per-realm basis. Note that this parameter honors the hierarchy of nested realms if you have them configured.

To enable bandwidth CAC for aggregate emergency sessions:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. If you are adding this feature to a pre-existing realm configuration, you will need to select and edit your realm.


```
ACMEPACKET(media-router)# realm-config
ACMEPACKET(realm-config)#
```
4. **max-priority-bandwidth**—Enter the amount of bandwidth you want to want to use for priority (emergency) calls. Note that the system first checks the **max-bandwidth** parameter, and allows the call if the value you set for priority calls is sufficient. If there is not enough priority and non-priority bandwidth allotted for an incoming call, the Net-Net SBC rejects it.

This parameter defaults to 0. You can enter any value between 0 and 999999999.

5. Save and activate your configuration.

Support for the AS Bandwidth Modifier

Two new parameters have been added to the media profile configuration:

- **sdp-bandwidth**—Enable or disable the use of the AS modifier in the SDP if the **req-bandwidth** and **sdp-rate-limit-headroom** parameters are not set to valid values in a corresponding media profile. The default value is **disabled**. The valid values are:
 - enabled | disabled
- **sdp-rate-limit-headroom**—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the **average-rate-limit** (rate limit for the RTP flow). The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—100

The following conditions apply to the use and application of these two new parameters:

- If the amount of required bandwidth is not specified in the media profile (**req-bandwidth**) for the media type in the **m=** line of the SDP, then the value specified in the AS modifier is used. The Net-Net SBC only uses the AS value if you set the new **sdp-bandwidth** to **enabled**.
- If the average rate limit value for RTP flows is not specified in the media profile (**average-rate-limit**) for the media type in the **m=** line of the SDP, then the value specified in the AS modifier is used. The Net-Net SBC only uses the AS value if you set the new **sdp-bandwidth** to **enabled**. When calculating the average rate limit that it will use based on the AS modifier, the Net-Net SBC applies the percentage set in the **sdp-rate-limit-headroom** parameter.
- The Net-Net SBC uses the value specified in the AS modifier (if **sdp-bandwidth** is enabled, and **req-bandwidth** is set to 0) along with the **user-cac-bandwidth** value set in the realm configuration; this works the same way that the **req-bandwidth** parameter does.
- The Net-Net SBC uses the value specified in the AS modifier (if **sdp-bandwidth** is enabled, and **req-bandwidth** is set to 0) along with the **max-bandwidth** value set in the realm configuration; this works the same way that the **req-bandwidth** parameter does.

ACLI Instructions and Examples: Setting the Media Type

To set any of the new media types in the media profile configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **media-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **media-type**—Enter the media type that you want to use for this media profile. The valid values are:
 - audio | video | application | data | image | text
5. Save and activate your configuration.

To set any of the new media types in the media policy configuration:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# media-policy
ACMEPACKET(media-policy)#
```
4. **media-type**—Enter the media type that you want to use for this media profile. The valid values are:
 - audio | video | application | data | image | text
5. Save and activate your configuration.

ACLI Instructions and Examples: Enabling AS Modifier Support and Headroom**To enable AS modifier use and establish the percentage of headroom to use:**

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **media-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **sdp-bandwidth**—Enable this parameter to use the AS bandwidth modifier in the SDP in the conditions described in the [Support for the AS Bandwidth Modifier \(666\)](#) section above. The default is **disabled**. Valid values are:
 - enabled | disabled
5. **sdp-rate-limit-headroom**—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the **average-rate-limit** (rate limit for the RTP flow). The default is **0**. The valid range is:
 - Minimum—0
 - Maximum—100
6. Save and activate your configuration.

About QoS Reporting

This section describes the Net-Net 9200 QoS reporting. QoS reporting provides you with real-time evaluation of network and route performance. It lets you contrast internal domain and external domain performance and facilitates SLA verification and traffic engineering. Net-Net 9200 QoS reporting is a measurement tool that collects statistics on Voice over IP (VoIP) call flows for SIP. To provide information, the Net-Net 9200 writes additional parameters to the Remote Authentication Dial-in User Service (RADIUS) call record. To provide information, the Net-Net SBC

writes additional parameters to the Remote Authentication Dial-in User Service (RADIUS) call record and Historical Data Recording (HDR) records.

You can use QoS statistics for SLA customer reporting, fault isolation, SLA verification, and traffic analysis. The Net-Net 9200 employs specialized hardware to inspect Real-Time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP) flows while maintaining wire-speed packet forwarding. QoS metrics are collected and reported on a per-session and per call-leg basis. These metrics are reported through real-time RADIUS records along with call accounting data.

Overview

When a conversation is established between two endpoints, two flows are present in each direction:

- RTP flow carries traffic between endpoints with a predictable packet arrival rate. The packets headers have sequence numbers that are used to determine whether packets are missing or lost.
- RTCP flow carries information about the RTP flow and keeps a different record. The RTCP packets contain timestamps based on Network Time Protocol (NTP).

QoS Statistics

Reported QoS data includes the following per-flow statistics:

- RTP and RTCP lost packets—Count of lost packets for both RTP and RTCP based on comparing the sequence numbers since the beginning of the call or the last context memory poll.
- RTP and RTCP average jitter—Incremental number of packets for both RTP and RTCP that have been used to generate the total and max jitter since the beginning of the call or the last context memory poll. The incremental accumulated jitter (in milliseconds) over all the packets received.
- RTP and RTCP maximum jitter—Maximum single jitter value (in milliseconds) for both RTP and RTCP from all the packets since the beginning of the call or the last context memory poll.
- RTCP average latency—Number of RTCP frames over which latency statistics have been accumulated and the incremental total of latency values reported since the beginning of the call or the last context memory poll.
- RTCP maximum latency—Highest latency value measured since the beginning of the call or the last context memory poll.
- RTP packet count
- RTP bytes sent and received
- RTCP lost packets—RTP lost packets reported in RTCP packets.
- ATP lost packets—Lost packets determined by monitoring RTP sequence numbers.

RADIUS Support

All the QoS statistics go into the RADIUS CDR. If a RADIUS client is configured on the Net-Net 9200, any time a call begins a START record is generated and sent. Only Stop RADIUS records contain the QoS statistic information.

For additional details about see the Net-Net system's support for RADIUS, including a dictionary of the standard RADIUS attributes supported by the Net-Net system and the Acme Packet accounting VSAs, see the *RADIUS Reference Guide*.

Configuring QoS

This section explains how to configure QoS. To generate QoS metrics, you need to enable QoS for the realm of the originating caller. The ingress realm determines whether QoS is turned on for a specific flow.

Note: If you run with QoS turned on one side only and disabled on the other you lose the ability to measure latency through the use of RTCP timestamps.

ACLI Instructions and Examples

To enable QoS:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Enable QoS for the realm.
 - 4a. **qos-enable**—Ensure this parameter is set to enabled. The default value is disabled.

The following example shows the **qos-enabled** parameter has been configured to enable QoS for this realm. All other **realm-config** parameters are deleted for brevity.

```
realm-config
    qos-enable                enabled
```

Network Management Controls

The Net-Net SBC supports network management controls for multimedia traffic specifically for static call gapping and 911 exemption handling. These controls limit the volume or rate of traffic for a specific set of dialed numbers or dialed number prefixes (destination codes).

In TDM networks, automatic call/code gapping was developed as part of the advanced intelligent network (AIN) to enable network element load shedding based on destination number (DN) in case of overload. However, since there are as yet no standards for call/code gapping for next generation multimedia networks, the Net-Net SBC provides statically-provisioned network management controls.

How It Works

To enable network management controls on your Net-Net SBC, you set up the ACLI **net-management-control** configuration and then enable the application of those rules on a per-realm basis. Each network management control rule has a unique name, in addition to information about the destination (IP address, FQDN, or destination number or prefix), how to perform network management (control type), whether to reject or divert the call, the next hop for routing, and information about status/cause codes. Details about the content of control rules as how to set them appear in the instructions and examples section.

When a SIP INVITE for a call arrives at the Net-Net SBC on an ingress realm where network management controls have been enabled, the Net-Net SBC takes the following steps:

- It searches the network management rules you have created (and which are stored in tables on the Net-Net SBC) for a control rule that best matches the newly-received call.
- If it does not find a matching control rule, the Net-Net SBC allows the call to proceed normally.
- If it finds a matching control rule, then the Net-Net SBC treats the call according to the specifics of the rule and the treatment method that it designates.

Matching a Call to a Control Rule

The Net-Net SBC uses the call classification key, specified by the **destination-identifier**, to match calls so that it can apply control rules. The call classification key specifies information about the destination, which can be an IP address, an FQDN, a destination (called) number, or destination prefix. You configure the classification key as part of the control rule.

Matching is performed from left to right, starting at the left-most character. A wildcard matches any digit.

The Net-Net SBC compares the following information from the SIP INVITE or H.323 Setup for matching:

- SIP INVITE—User part of the Request URI, or the host part of the Request URI
- H.323 Setup—Q.931 Called Party Number IE

The Net-Net SBC normalizes the user-part of the Request-URI prior to performing any matching for NMC based on the dialed number. This normalization strips out any of the visual-separator characters. For example:

- +358-555-1234567 after normalization is: +3585551234567
- +(358)555.1234567 after normalization is: +3585551234567

Normalization occurs only for NMC look-up purposes, and it does not alter the actual Request-URI. If multiple rules match an incoming call, then the Net-Net SBC gives first priority to destination number or the destination prefix. Next, it tries to match based on the IP address, and finally it looks to the domain (lowest priority).

Specifically, the Net-Net SBC supports the following:

- The user-part can contain escaped sequences that the Net-Net SBC normalizes to their unescaped representation. For example:
 - %23(358)555.1234567 after normalization is: #3585551234567.
- The Net-Net SBC parses the user-part of the Request-URI up to the first semicolon (;). For example, the user-part in tel:+358-555-1234567;postd=pp22 will be +358-555-12134567.
- You can use the caret (^) to wildcard digits 0-9 in the destination-identifier you configure.
- You can enter and match special characters such as the plus-sign (+), the asterisk (*), the pound sign (#), capital letter A (A), capital letter B (B), capital

letter C (C), capital letter D (D), capital letter E (E), capital letter F (F), lowercase letter p (p), lowercase letter w (w).

For IWF Calls

For calls that require interworking between SIP and H.323, the Net-Net SBC performs call management control on the ingress leg of the call. If the call matches a control rule, the Net-Net SBC applies the treatment rule at the time it processes the ingress call. In addition:

- When the treatment method is rejection, the Net-Net SBC immediately rejects the call on the ingress leg.
- When the control rule type is priority, the Net-Net SBC's SIP or H.323 process adds a request URI parameter that flags the call as priority before it sends an INVITE to either the SIP or H.323 process. When the INVITE is received and the priority request URI is found, the Net-Net SBC makes the call and bypasses any local network management.

Call Handling Determination

There are three types of control rules from which you can choose; each is a different way for the Net-Net SBC to handle calls matching the classification key:

- Call gap rate—Controls the maximum sustained rate of calls that match the classification key.

Using this type, the Net-Net SBC calculates the time since the last matching call. If that time is equal to or greater than the minimum time gap set in the control rule (i.e., it does not exceed the rate), then the call proceeds normally. If the call is less than the minimum time gap (i.e., it causes the call rate to be exceeded), then the Net-Net either rejects or diverts the call.

To keep the call rate below the control value, the Net-Net SBC ensures a minimum call gap time between the matching calls. For example, if the control value is 10 calls per second, the minimum call gap time would be 0.1 second. And if a matching call were to arrive within a tenth of a second since the last matching call, then the Net-Net SBC applies the treatment method.

- Call gap percentage—Controls the percentage of calls matching the classification key you set for the control rule.

When using this control rule type, the Net-Net SBC applies the treatment method to the percentage of matching calls (that you set in the value parameter) out the total number of matching calls it receives. For example, if you set the value parameter for the control rule to 50 and use this control type, the Net-Net SBC applies the treatment method to every other call it receives (or 50% of the calls it receives) that matches the classification key.

Note that the Net-Net SBC cannot maintain exact percentages for the control value at all times, especially at system start-up when the number of incoming calls is small.

- Priority—Exempts calls to a destination (like 911) from local network management controls such as:
 - Session agent constraints
 - Bandwidth constraints (such as per-realm bandwidth)
 - External policy servers (requests are made to the policy server; calls are admitted and processed regardless of the decision or reachability of the policy server)
 - Per-user call admission control

- CPU constraints

The Net-Net SBC will not bypass licensing constraints, however.

Treatment Methods

You can choose from two different treatment methods:

- Call rejection—The Net-Net SBC rejects the call.
 - For SIP, the Net-Net SBC sends a response messages with the status code of the control rule. This response message also includes a Reason header with the Q.850 cause code that you configure as part of the control rule; it contains only the Q.850 cause code, and there is no reason-text included. For example:

Reason: Q.850; cause=63

- Call diversion—The Net-Net SBC routes the call to the location you specify in the control rule's next hop parameter.

Except for this routing, the call proceeds as normal. Local treatments such as number translation apply to the call, as do local controls such as licensing. Note the following:

- If the next hop is an FQDN, the Net-Net SBC performs DNS queries to resolve the next hop to an IP address so that it can route the call properly.
- If the next hop is a session agent group, the Net-Net SBC selects a session agent from the group according to the selection strategy you set for the group. Then the Net-Net SBC uses the IP address of the selected session agent.

ACLI Instructions and Examples

In order use the network management controls feature, you need to set control rules and then enable their application on a per-realm basis. This section shows you how to set up those configuration.

Configuring an Individual Control Rule

To configure individual network management control rule:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
```
4. **name**—Enter the name of this network management control rule; this value uniquely identifies the control rule. There is no default for this parameter.
5. **state**—Leave this parameter set to **enabled** (default), or change it to **disabled** to **disabled** this control rule.
6. **destination-identifier**—Enter the classification key. This parameter specifies information about the destination, which can be an IP address, an FQDN, a destination (called) number, or destination prefix. You can wildcard characters in the classification key using the caret symbol (^).

This parameter can accommodate a list of entries so that, if necessary, you can specify multiple classification keys. You can edit the list of classification keys

using the ACLI **add-destination-identifier** and **remove-destination-identifier** commands from within the network management controls configuration.

7. **type**—Enter the control type that you want to use, or leave this parameter set to its default: **NONE**. Your choices are outlined in the table below:

Control Type	Description
NONE	No control type is applied.
GAP-RATE	Controls the maximum sustained rate of calls that match the classification key.
GAP-PERCENT	Controls the percentage of calls matching the classification key you set for the control rule.
PRIORITY	Exempts calls to a destination (like 911) from local network management controls.

For more information about control types, refer to the [Call Handling Determination \(671\)](#) section above.

8. **value**—When you set the control type to either **GAP-RATE** or **GAP-PERCENT**, enter the maximum sustained rate of calls or the percentage of calls to which you want the control rule applied. The default is zero, and:
- For the **GAP-RATE**, the maximum is 2147483647 (which you can set by entering -1 as the value, an option provided for ease of use)
 - Using the minimum value (0) means that the Net-Net SBC treats all calls
 - Using the maximum value means that the Net-Net SBC treats no calls
 - For the **GAP-PERCENT**, the maximum is 100
 - Using the minimum value (0) means that the Net-Net SBC treats no calls
 - Using the maximum value (100%) means that the Net-Net SBC treats all calls
9. **treatment**—Enter the treatment method that your want to use or leave this parameter set to **NONE** (default):

Control Type	Description
NONE	No treatment is applied.
REJECT	The Net-Net SBC rejects the call.
DIVERT	The Net-Net SBC routes the call to the location you specify in the control rule's next hop parameter.

For more information about control types, refer to the [Treatment Methods \(672\)](#) section above.

10. **next-hop**—Enter the next hop for the Net-Net SBC to use when the treatment method is **DIVERT**. This value should contain one of the following:
- hostname(:port)
 - IP address(:port)

- Name of a valid, configured session agent
 - Name of a valid, configured session agent group—When you set this parameter to a session agent group, you must specify that it is a session agent group by prepending the name of the group with either **SAG:** or **sag:**. For example, the entry for a session agent group with **Group2** as its name would be **SAG:Group2** or **sag:Group2**.
11. **realm-next-hop**—Enter the realm identifier to designate the realm of the next hop when the treatment type is **DIVERT**.
 12. **protocol-next-hop**—Enter the signaling protocol for the next hop when the treatment type is **DIVERT**.
 13. **status-code**—Enter the SIP response code that you want the Net-Net SBC to use when the treatment method is **REJECT**, or leave this parameter set to its default, **503** (Service Unavailable). The value you enter must be numeric, but it cannot be a 0.
 14. **cause-code**—Enter the Q.850 cause code that you want the Net-Net SBC to use when the treatment method is **REJECT**. You can leave this parameter set to its default, **63** (Service or option not available) or set it to any numeric value except 0.
 - For a SIP call, the Net-Net SBC replaces the cause code in the Reason header of the SIP response.

Applying a Network Management Control Rule to a Realm

Once you have configured network management control rules, you can enable their use on a per-realm basis.

To apply a network management control rule to a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```

If you are enabling network management controls for a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **net-management-control**—Set this parameter to **enabled** to apply network control rules in this realm. The default value is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

Management

This section explains how to view and reset network management control statistics, and it contains SNMP information.

Displaying Network Management Control Statistics

You can use the new ACLI **show net-management-control** command to see the statistics that the Net-Net SBC collects. When you use the command, you specify the name of the network management control rule for which you want to display data or you can enter **all** to see the statistics for all control rules.

For each network management control rule, the Net-Net SBC gathers statistics for the number of:

- Incoming calls—Incoming calls that match the destination identifier
- Rejected calls—Calls that were rejected as a result of the control rule being applied
- Diverted calls—Incoming that were diverted as a result of the control rule being applied

The display you see when you execute this command shows statistics for the current period, lifetime, and maximum value in a period.

To display network management control statistics:

1. In either User or Superuser mode, type the **show net-management-control** command, a <Space>, and then the name of the control rule for which you want to see data. You can enter **all** if you want to see the data for all control rules. Then press <Enter>.

```
ACMEPACKET# show net-management-control nmcpercent
14:45:15-63
Name: nmcpercent
Type: gap-percent
----- Lifetime -----
Current      Total      PerMax
Incoming Calls      0          0          0
Rejected Calls      0          0          0
Diverted Calls      0          0          0
```

Resetting Network Management Control Statistics

To reset network management control statistics, you use the ACLI **reset net-management-control** command followed by the name of the control rule for which you want to reset statistics. This command resets the counters to zero (0).

To reset network management control statistics:

1. In Superuser mode, type the ACLI **reset net-management-control** command, a <Space>, and then the name of the control rule for which you want to see data. Then press <Enter>.

```
ACMEPACKET# reset net-management-control nmcpercent
```

SNMP Information

You can also retrieve network management control information with an SNMP GET. The MIB for network management controls is as follows:

1.3.6.1.4.1.9148.3.2.1.3	apSysMgmtMIBNetMgmtCtrlObjects	NODE	
1.3.6.1.4.1.9148.3.2.1.3.1	apNetMgmtCtrlStatsTable	NODE	
1.3.6.1.4.1.9148.3.2.1.3.1.1	apNetMgmtCtrlStatsEntry	NODE	
1.3.6.1.4.1.9148.3.2.1.3.1.1.1	apNetMgmtCtrlStatsName	LEAF	DisplayString
1.3.6.1.4.1.9148.3.2.1.3.1.1.2	apNetMgmtCtrlStatsType	LEAF	ApNetMgmtCtrlType
1.3.6.1.4.1.9148.3.2.1.3.1.1.3	apNetMgmtCtrlStatsIncomingTotal	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.4	apNetMgmtCtrlStatsRejectedTotal	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.5	apNetMgmtCtrlStatsDivertedTotal	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.6	apNetMgmtCtrlStatsIncomingCurrent	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.7	apNetMgmtCtrlStatsRejectedCurrent	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.8	apNetMgmtCtrlStatsDivertedCurrent	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.9	apNetMgmtCtrlStatsIncomingPeriodMax	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.10	apNetMgmtCtrlStatsRejectedPeriodMax	LEAF	Unsigned32
1.3.6.1.4.1.9148.3.2.1.3.1.1.11	apNetMgmtCtrlStatsDivertedPeriodMax	LEAF	Unsigned32

Trap Name	Description
ApNetMgmtCtrlStatsType	Type of the specified network management control. Options for its value are: gap-rate, gap-percent, or priority.
ApNetMgmtCtrlStatsIncomingTotal	Total number of incoming calls that match a destination identifier of the network management control rule. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsRejectedTotal	Number of ApNetMgmtCtrlStatsIncomingTotal that are rejected. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsDivertedTotal	Number of ApNetMgmtCtrlStatsIncomingTotal that are diverted. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsIncomingCurrent	Number of incoming calls during the current period that match a destination identifier of the network management control rule. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsRejectedCurrent	Number of ApNetMgmtCtrlStatsIncomingCurrent that are rejected. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsDivertedCurrent	Number of ApNetMgmtCtrlStatsIncomingCurrent that are diverted. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsIncomingPeriodMax	Maximum number of incoming calls during a period that match a destination identifier of the net management control rule. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsRejectedPeriodMax	Number of ApNetMgmtCtrlStatsIncomingPeriodMax that are rejected. This statistic can be reset via the ACLI command reset net-management-control command.
ApNetMgmtCtrlStatsDivertedPeriodMax	Number of ApNetMgmtCtrlStatsIncomingPeriodMax that are diverted. This statistic can be reset via the ACLI command reset net-management-control command.

NMC Call Gapping Enhancements

The Net-Net SBC's network management controls (NMC) feature is now enhanced to provide greater flexibility, and allow a desired call-per-second (CPS) threshold to be achieved or surpassed by a predictable amount. Using this feature, you can enable the Net-Net SBC to average the call rate, and widen the period of a surge that would invoke NMC rules.

In prior releases, the NMC carries out a call gapping policy that monitors the arrival times between INVITEs, and then compares the arrival times to with the threshold. To enable this, you set the **type** parameter to **gap-rate**, and then configure the **value** parameter with the maximum sustained rate of calls. The threshold is equal to 1/gap-rate value. However, this implementation means that if two calls arrive simultaneously at the Net-Net SBC, one of them might be rejected or diverted if it exceeds the threshold and the control rule is applied. This is the case even when the sustained call rate does not exceed the control rule.

To resolve this, call gapping has been refined to support two new parameters that form part of an calculation the Net-Net SBC performs for applying NMC rules. Using the current time, the time of the last call gapped, the call counter value (tracked internally by the Net-Net SBC), the CPS value for the gap-rate control rule, and the values of the new parameters, the Net-Net SBC performs calculations that determine whether or not to apply the control rule.

How It Works

The Net-Net SBC employs this leaky bucket algorithm to enforce calls per second. It smooths the call rate over a defined window of time to protect against surges. The values used for the calculation are:

- A—Calls per second; configure by setting the **type** parameter to **gap-rate**, and the **value** parameter to the CPS you want enforced
- m—Maximum counter value; must be greater than 0
- W—Window size; must be greater than
- deltaT—Time between allowed calls matching an NM control rule

The calculation is performed as follows, with the noted results:

- $1 + m - m * A * \text{deltaT} / W \leq M$ —Means the call is allowed
- $1 + m - m * A * \text{deltaT} / W > M$ —Means that NMC rules are applied

Note the following:

- Setting the counter value and the window size to the same values guarantees that the processed CPS load will not exceed the desired CPS target.
- As the counter value becomes greater than the window size value, rejection rate will drop and the desired CPS threshold is not guaranteed.
- Increasing the window size results in a lower rejection rate when the attempted CPS is the same as the desired CPS; as the attempted CPS rate increases, rejection rates increase at a steeper rate.
- If either the count rate or the window size is set to 0, then the Net-Net SBC reverts to call gapping behavior from prior releases.

ACLI Instructions and Examples

NMC call gapping adds two new configuration parameters to the network management controls configuration:

- **gap-rate-max-count**—Maximum count that triggers the application of network management control rule if it is exceeded.
- **gap-rate-window-size**—Length of time in seconds used for the gapping rate calculation.

For this feature to behave as intended, you also need to set the **type** parameter to **gap-rate**, and set the **value** parameter to the maximum sustained rate of calls that you want to support.

To configure NMC call gapping enhancements:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-related configurations.


```
ACMEPACKET(configure)# session-router
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
```

To add support to a pre-existing network management control configuration, use the ACLI **select** command to choose the configuration you want to edit.
4. **gap-rate-max-count**—Maximum count that triggers the application of network management control rule if it is exceeded. The default is 0.

Along with the current time, the last time of a gapped call, the call counter value, the CPS value, and the gap-rate-window-size value, the Net-Net SBC uses **gap-rate-max-count** as a measurement to determine if a control rule will be applied.
5. **gap-rate-window-size**—Length of time in seconds used for the gapping rate calculation. The default is 0.

Along with the current time, the last time of a gapped call, the call counter value, and the CPS value, the Net-Net SBC uses the **gap-rate-window-size** value to calculate whether the maximum count is within allowable limits.
6. Save and activate your configuration.

Configuring the Account Configuration and Account Servers

This section explains how to configure the account configuration and account servers so you can use the Net-Net system in conjunction with external RADIUS (accounting) servers to generate CDRs and provide billing services requires.

For more information about RADIUS, see the *Net-Net RADIUS Reference Guide*.

ACLI Instructions and Examples

To configure the account configuration and account servers:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **account-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# account-config
ACMEPACKET(account-config)#
```
4. To configure account server parameters (a subset of the account configuration parameters, type **account-servers** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(account-config)# account-servers
```

```
ACMEPACKET(account-server)#
```

The following example shows both the account config and account server parameters.

```

account-config
  hostname                acctserver1
  port                    1813
  strategy                Hunt
  state                  enabled
  max-msg-delay           60
  max-wait-failover       100
  trans-at-close          disabled
  generate-start          OK
  generate-interim        OK
                          Reinvite-Response
account-server
  hostname                packet.com
  port                    1813
  state                  enabled
  min-round-trip          100
  max-inactivity          100
  restart-delay           100
  bundle-vsa              enabled
  secret                  testing
  NAS-ID                  acme-accounting
last-modified-date        2005-01-15 02:23:42

```

Account Configuration

You set the account configuration parameters to indicate where you want accounting messages sent, when accounting messages you want them sent, and the strategy you want used to select account servers.

To configure the account configuration:

1. **hostname**—Enter a name for the host associated with the Net-Net system in hostname (FQDN) format. The default value is the name of the local host.
The value you enter here must match the configured physical interface's operation type **control** or **maintenance**, to determine on which network to send RADIUS messages.
2. **port**—Retain the default value of 1813 or enter the number of the UDP port associated with the Net-Net system from which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535

3. **strategy**—Indicate the strategy you want used to select the accounting servers to which the Net-Net system will send its accounting messages. The following table lists the available strategies:

Strategy	Description
hunt	Selects accounting servers in the order in which they are listed. If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers
failover	Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
round robin	Selects each accounting server in order, distributing the selection of each accounting server evenly over time.
fastest round trip time	Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
fewest pending	Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the Net-Net system).

4. **state**—Retain the default value **enabled** if you want the account configuration active on the system. Enter **disabled** if you do not want the account configuration active on the system.
5. **max-msg-delay**—Retain the default value of **60** seconds or indicate the length of time in seconds that you want the Net-Net system to continue trying to send each accounting message. During this delay, the Net-Net system can hold a generic queue of 4096 messages.
- minimum: zero (0)
 - maximum: $2^{32}-1$
6. **max-wait-failover**—Retain the default value of **100** messages or indicate the maximum number of accounting messages the Net-Net system can store its message waiting queue for a specific accounting server, before it is considered a failover situation.
- Once this value is exceeded, the Net-Net system attempts to send its accounting messages, including its pending messages, to the next accounting server in its configured list.
- minimum: one (1) message
 - maximum: 4096 messages
7. **trans-at-close**—Retain the default value of **disabled** if you do not want to defer the transmission of message information to the close of a session. Enter **enabled** if you want to defer message transmission.
- **disabled**: the Net-Net system transmits accounting information at the start of a session (Start), during the session (Interim), and at the close of a session

(Stop). The transmitted accounting information for a single session might span a period of hours and be spread out among different storage files.

- **enabled:** limits the number of files on the Net-Net system used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.
8. **generate-start**—Retain the default value **ok** if you want the RADIUS Start message to be generated once the Net-Net system receives an OK message in response to an INVITE. (A RADIUS Start message informs the accounting server that a SIP session has started.)

Other options include:

- **start:** RADIUS Start message should not be generated
 - **invite:** RADIUS Start message should be generated once the Net-Net system receives a SIP session INVITE.
9. **generate-interim**—Retain the default value **reinvite response** to cause the Net-Net system to transmit a RADIUS Interim message. (A RADIUS Interim message indicates to the accounting server that the SIP session parameters have changed.)

You can select none, one, or more than one of the following values:

Option	Description
ok	RADIUS Start message is generated when the Net-Net system receives an OK message in response to an INVITE.
reinvite	RADIUS Interim message is generated when the Net-Net system receives a SIP session reINVITE message.
reinvite response (default)	RADIUS Interim message is generated when the Net-Net system receives a SIP session reINVITE and responds to it (for example, session connection or failure).
reinvite cancel	RADIUS Interim message is generated when the Net-Net system receives a SIP session reINVITE, and the Reinvite is cancelled before the Net-Net system responds to it.

10. **account-server**—Create the account server list to store accounting server information for the account configuration. Each account server can hold 100 accounting messages. See the next section for step-by-step instructions.

Account server entries are specific to the account configuration. They cannot be viewed or accessed for editing outside of the account configuration.

Note: RADIUS will not work if you do not enter one or more servers in a list.

Account Server

You must establish the list of servers to which the Net-Net system can send accounting messages.

1. **hostname**—Name of the host associated with the account server in hostname format (FQDN) or as an IP address.
2. **port**—Retain the default **1813** or enter the number of the UDP port associated with the account server to which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535
3. **state**—Retain the default **enabled** to enable the account servers on the system or enter **disabled** to disable them.
4. **min-round-trip**—Retain the default **250** milliseconds or indicate the minimum round trip time of an accounting message.
 - minimum: 1025 milliseconds
 - maximum: 65535 milliseconds

A round trip consists of the following:

- The Net-Net system sends an accounting message to the account server.
- The account server processes this message and responds back to the Net-Net system.

If the fastest RTT is the strategy for the account configuration, the value you enter here can be used to determine an order of preference (if all the configured account servers are responding in less than their minimum RTT).

5. **max-inactivity**—Retain the default **60** seconds or indicate the length of time in seconds that you want the Net-Net system with pending accounting messages to wait when it has not received a valid response from the target account server.
 - minimum: 1 second
 - maximum: 300 seconds

Once this timer value is exceeded, the Net-Net system marks the unresponsive account server as disabled in its failover scheme. When a server connection is marked as inactive, the Net-Net system attempts to restart the connection and transfers pending messages to another queue for transmission. RADIUS messages might be moved between different account servers as servers become inactive or disabled.

6. **restart-delay**—Retain the default **30** seconds or indicate the length of time in seconds you want the Net-Net system to wait before resending messages to a disabled account server.
 - minimum: 1 second
 - maximum: 300 seconds

7. **bundle-vsa**—Retain the default **enabled** if you want the account server to bundle the VSAs within RADIUS accounting messages. Enter **disabled** if you do not want the VSAs to be bundled. (Bundling means including multiple VSAs within the vendor value portion of the message.)

In a bundled accounting message, the RADIUS message type is vendor-specific, the length is determined for each individual message, and the vendor portion begins with a 4-byte identifier, and includes multiple vendor type, vendor length, and vendor value attributes.

8. **secret**—Enter the secret passed from the account server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IPv4 address of the received packet.
9. **NAS-ID**—Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the Net-Net system for the transmittal of accounting messages.

The remote server to which the account configuration sends messages uses at least one of two potential pieces of information for purposes of identification. The Net-Net system accounting messages always includes in the first of these:

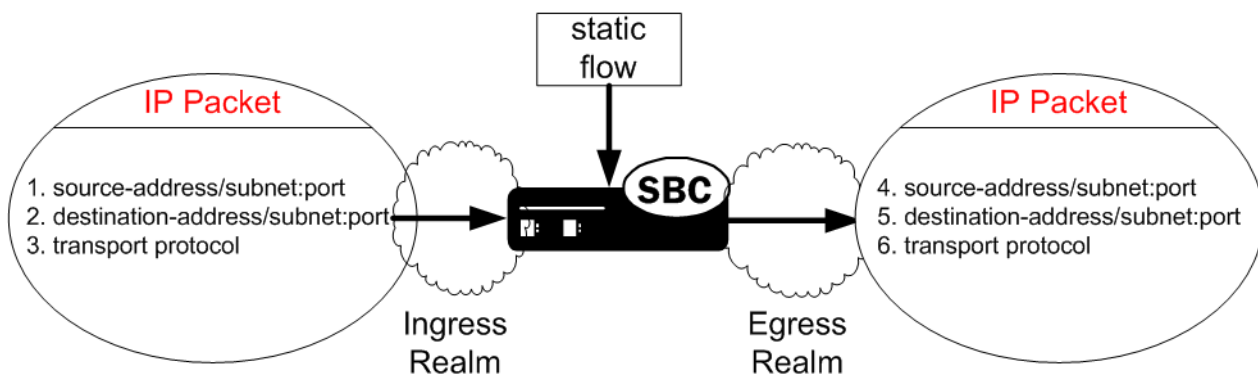
- Network Access Server (NAS) IP address (the IP address of the Net-Net system's SIP proxy)
- NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

Introduction

This chapter describes the Net-Net 9200's static flows feature. Static flows allow network traffic that matches specific criteria to pass through the Net-Net 9200 unrestricted. This feature lets you steer traffic toward a particular destination based on its original characteristics. Static flows can range from being widely accessible to very restrictive, depending on the values you establish. Static flows are used for transporting a variety of signaling messages through the Net-Net 9200 to achieve vendor interoperability.

About Static Flows

The static flow element explicitly writes entries into the IP routing table. These entries are persistent and are not deleted as calls are set up and broken down. Refer to the following diagram to understand how a static flow works.



A static flow entry watches for traffic with specific criteria on a specified ingress realm; that traffic consists of the following criteria:

1. The IPv4 packet enters the Net-Net 9200 on the specified ingress realm.
2. The packet contains matching source address, subnet, and port criteria, field 1.
3. The packet contains matching destination address, subnet, and port criteria, field 2.
4. The packet contains a matching transport protocol, field 3.

If the above conditions are met, then the Net-Net 9200 does the following:

1. The IPv4 traffic is forwarded out of the Net-Net 9200 on the specified egress realm.
2. The configured source address, subnet, and port criteria are written to the exiting packet, field 4.
3. The configured destination address, subnet, and port criteria are written to the exiting packet, field 5.

4. The original transport protocol and its contents remain unchanged as the packet exits into the egress realm.

About Network Address Translation ALG

The Net-Net 9200 supports Network Address and Port Translation (NAPT) and Trivial File Transfer Protocol (TFTP) functionality over media interfaces, collectively known as Network Address Translation (NAT) ALG. The NAT ALG feature is implemented as an extension of the static flow feature.

In some applications, the Net-Net 9200 acts as an intermediary device, positioned between endpoints located in an access network and application servers located in a backbone network. The Net-Net 9200's NAT ALG feature enables these endpoints to use non-VoIP protocols, such as TFTP and HTTP, to access servers in a provider's backbone network to obtain configuration information.

NAT ALG parameters support RTC and can be dynamically reconfigured. The active NAT ALG configuration can be replicated on the standby 9200 in an HA configuration.

NAPT

The NAPT ALG functionality is the same as that found in commercially available enterprise and residential NAT devices. The Net-Net 9200 watches for packets entering a media interface that match source and destination IP address criteria. Matching packets are then redirected out of the egress interface, through a specified port range, toward a destination address.

TFTP

The TFTP ALG is implemented as an extension of the NAT ALG. It works slightly differently than traditional NAPT. In a TFTP session, the first packet is sent from a source endpoint to port 69 on the TFTP server. The TFTP server responds from another port. This port, from which the TFTP response originates, is used for the remainder of the TFTP session.

To act as a TFTP ALG, the Net-Net 9200 will latch on the first return packet from the server to learn the server's port. The ingress-side destination port of the Net-Net 9200 is changed to reflect the new communications port for the TFTP session. This process takes place without any user intervention.

Configuring Static Flows

This section explains how to configure static flows. It also provides sample configurations for your reference. You can configure static flows with or without NAT ALG. If you configure static flows with NAT ALG, you can choose NAPT or TFTP as the ALG type.

Basic Static Flow Configuration Overview

This section outlines the basic static flow configuration, without NAT ALG. You configure static flows by specifying ingress traffic criteria followed by egress re-sourcing criteria.

When configuring static flows, the following conventions are used:

- An IPv4 address of 0.0.0.0 matches all IPv4 addresses.
 - Not specifying a port implies all ports.
 - Not specifying a subnet mask implies a /32, matching for all 32 bits of the IPv4 address.
1. Set the static flows' incoming traffic-matching criteria. First set the ingress realm where you expect to receive traffic that will be routed via a static flow. Second, set the traffic's source IPv4 address, source subnet, and source port or port range criteria. Third, set the traffic's destination IPv4 address, destination subnet, and destination port criteria. This is usually an external address on the Net-Net 9200.
 2. Set the criteria that describes how traffic should be translated on the egress side of the Net-Net 9200. First set the egress realm where you want to send the traffic to be routed by this static flow. Second, set the traffic's source IPv4 address, source subnet, and source port or port range criteria. This is usually an external address on the Net-Net 9200. Third, set the traffic's destination IPv4 address, destination subnet, and destination port criteria.
 3. Set the protocol this static flow entry acts upon. This type of packet, as the payload of the IPv4 packet, remains untouched as traffic leaves the Net-Net 9200. Specifying a layer 4 protocol here acts as another criteria to filter against for this static flow.

The combination of entries in the ingress realm, ingress source address, ingress destination address, and protocol fields must be unique. For bidirectional traffic, you need to define a separate static flow in the opposite direction.

ACLI Instructions and Examples

This section describes how to configure the **static-flow** element using the ACLI.

About the Static Flow Parameters

The ingress IP address criteria is set first. These parameters are applicable to traffic entering the ingress side of the Net-Net 9200.

- **in-realm-id**—The access realm, where endpoints are located.
- **in-source**—The source network in the access realm where the endpoints exist. This parameter is entered as an IP address and netmask in slash notation to indicate a range of possible IP addresses.
- **in-destination**—The IP address and port pair where the endpoints send their traffic. This is usually the IP address and port on a Net-Net 9200 physical interface that faces the access realm.

The egress IP address criteria is entered next. These parameters determine how traffic is re-sourced as it leaves the Net-Net 9200 and enters the backbone network.

- **out-realm-id**—The backbone realm, where servers are located.
- **out-source**—The IP address on the physical interface of the Net-Net 9200 where traffic exits the Net-Net 9200 into the backbone realm. Do not enter a port for this parameter.
- **out-destination**—The IP address and port pair destination of the traffic. This is usually a server in the backbone realm.
- **protocol**—The protocol associated with the static flow. The protocol you choose must match the protocol in the IPv4 header. Valid entries are TCP, UDP, ICMP, ALL.

The type of NAT ALG, if any.

- **alg-type**—The type of NAT ALG. Set this to NAPT, TFTP, or none.

The port range for port re-sourcing as traffic affected by the NAT ALG exits the egress side of the Net-Net 9200 is set next. (Not applicable if **alg-type** is set to none.)

- **start-port**—The starting port the NAT ALG uses as it re-sources traffic on the egress side of the Net-Net 9200.
- **end-port**—The ending port the NAT ALG uses as it re-sources traffic on the egress side of the Net-Net 9200.

The flow timers are set next. (Not applicable if **alg-type** is set to none.)

- **flow-time-limit**—Total session time limit in seconds. The default is 0; no limit.
- **initial-guard-timer**—Initial flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit.
- **susbsq-guard-timer**—Subsequent flow guard timer for an ALG dynamic flow in seconds. The default is 0; no limit.

Finally, you can set the optional bandwidth policing parameter for static flows (with or without NAT ALG applied).

- **average-rate-limit**—Sustained rate limit in bytes per second for the static flow and any dynamic ALG flows. The default is 0; no limit.

Configuring Static Flow

To configure static flow:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the **media-manager** path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **static-flow** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

ACMEPACKET(media-manager)# **static-flow**

From this point, you can configure media policing parameters.

4. **in-realm-id**—Set the ingress realm to be watched for traffic to translate. This parameter must match a realm ID already configured in the **realm-config** element.
5. **description**—Enter an optional description for this static flow. If your entry contains any spaces, you must put the entire entry in double quotation marks.
6. **in-source**—Enter the network that defines where the incoming traffic originates. This parameter is entered as an IPv4 address and netmask in slash notation.
7. **in-destination**—Enter the ingress side's destination. This is the IPv4 address and port of the network interface in the ingress realm.
8. **out-realm-id**—Enter the defined realm where traffic leaving this NAT ALG exits the Net-Net 9200.
9. **out-source**—Enter the egress IPv4 address. This is the IPv4 address of the network interface where traffic subject to the NAT ALG you are defining leaves the Net-Net 9200. Do not enter a port number for this parameter.
10. **out-destination**—Enter the IPv4 address and port number of the server or other destination to which traffic is directed.
11. **protocol**—Enter the protocol this NAT ALG acts upon. Valid values are: TCP, UDP, ICMP, ALL.
12. **alg-type**—Enter the ALG type. Valid values are: none, NATP, TFTP.
13. **start-port**—Enter the beginning port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this NATP ALG redirects.
14. **end-port**—Enter the ending port number of the port range that the Net-Net 9200 allocates on the egress side for flows that this NATP ALG redirects.
15. **flow-time-limit**—Enter the total time limit for a flow in seconds. The default is 0; no limit.
16. **initial-guard-timer**—Enter the initial guard timer value in seconds. The default is 0; no limit.
17. **subsq-guard-timer**—Enter the subsequent guard timer value in seconds. The default is 0; no limit.
18. **average-rate-limit**—Enter a maximum sustained rate limit in bytes per second. The default is 0; no limit.

The following example shows a **static-flow** configuration element configured for a NATP ALG.

in-realm-id	access
description	static flow 1
in-source	172.16.0.0/16
in-destination	172.16.1.16:23
out-realm-id	backbone
out-source	172.168.24.16
out-destination	172.168.24.95:23
protocol	UDP
alg-type	none
start-port	0
end-port	0
flow-time-limit	0

initial-guard-timer	60
subsq-guard-timer	60
average-rate-limit	0

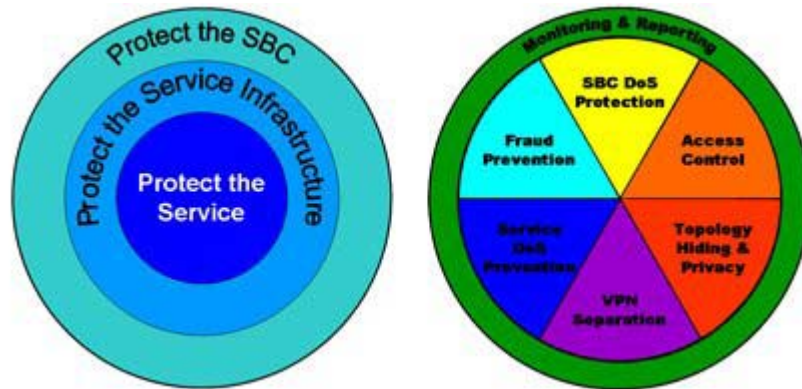
Introduction

This chapter explains Net-Net 9200 security, which is designed to provide security for VoIP and other multimedia services. It includes access control, DoS attack, and overload protection, which help secure service and protect the network infrastructure (including the Net-Net 9200). In addition, Net-Net 9200 security lets legitimate users still place calls during attack conditions, thereby protecting the service itself.

Security Overview

Net-Net 9200 security includes the Net-SAFE framework's numerous features and architecture designs. Net-SAFE is a requirements framework for the components required to provide protection for the Session Border Controller (SBC), the service provider's infrastructure equipment (proxies, gateways, call agents, application servers, and so on), and the service itself.

The following diagrams illustrate Net-SAFE:



Each of Net-SAFE's seven functions consists of a collection of more specific features:

- Session border controller DoS protection—Autonomic, SBC self-protection against malicious and non-malicious DoS attacks and overloads at Layers 2 to 4 (TCP, SYN, ICMP, fragments, and so on) and Layers 5 to 7 (SIP signaling floods, malformed messages, and so on).
- Access control—Session-aware access control for signaling and media using static and dynamic permit/deny access control lists (ACLs) at layer 3 and 5.
- Topology hiding and privacy—Complete infrastructure topology hiding at all protocol layers for confidentiality and attack prevention security. Also, modification, removal or insertion of call signaling application headers and fields. Includes support for the SIP Privacy RFC.

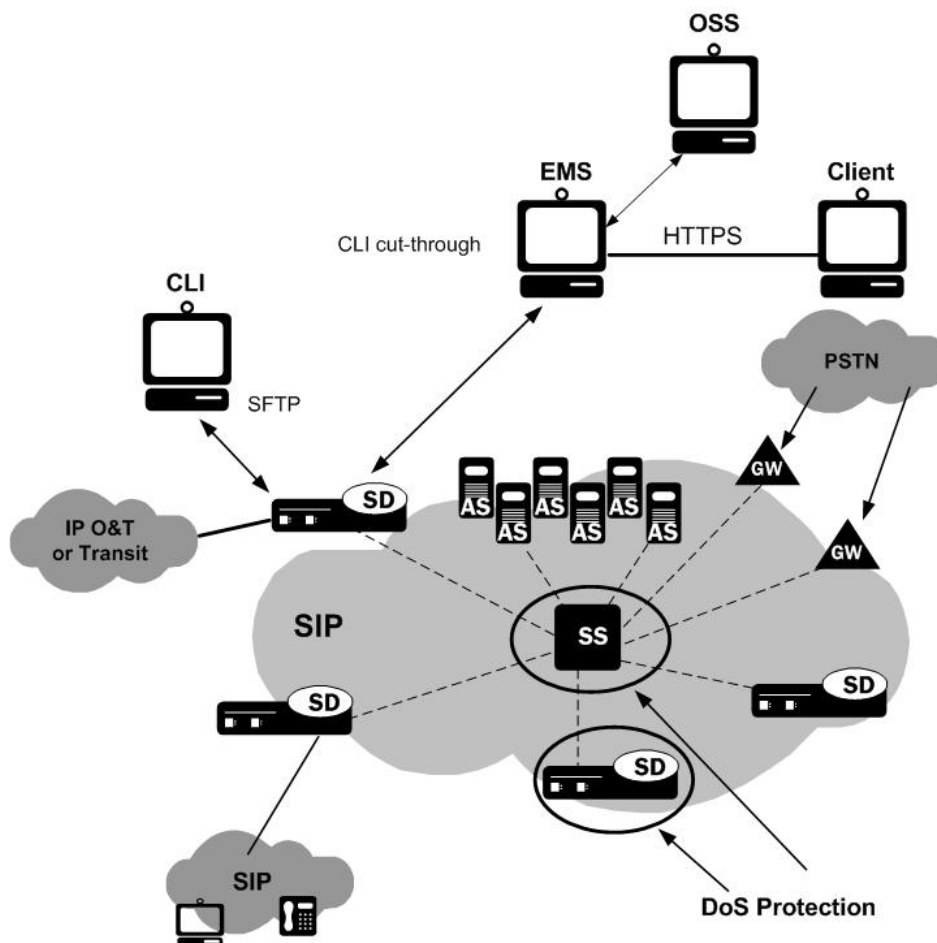
- VPN separation—Support for Virtual Private Networks (VPNs) with full inter-VPN topology hiding and separation, ability to create separate signaling and media-only VPNs, and with optional intra-VPN media hair-pinning to monitor calls within a VPN.
- Service infrastructure DoS prevention—Per-device signaling and media overload control, with deep packet inspection and call rate control to prevent DoS attacks from reaching service infrastructure such as SIP servers, softswitches, application servers, media servers or media gateways.
- Fraud prevention—Session-based authentication, authorization, and contract enforcement for signaling and media; and service theft protection.
- Monitoring and reporting—Audit trails, event logs, access violation logs and traps, management access command recording, Call Detail Records (CDRs) with media performance monitoring, raw packet capture ability and lawful intercept capability. The monitoring method itself is also secured, through the use of SSH and SFTP, and through the ability to use a separate physical Ethernet port for management access.

Denial of Service Protection

The Net-Net 9200 DoS protection functionality protects softswitches and gateways with overload protection, dynamic and static access control, and trusted device classification and separation at Layers 3-5. The Net-Net 9200 itself is protected from signaling and media overload, but more importantly the feature allows legitimate, trusted devices to continue receiving service even during an attack. DoS protection prevents the Net-Net 9200 host processors from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Rate-based attack (flood) of valid or invalid call requests, signaling messages, and so on
- Overload of valid or invalid call requests from legitimate, trusted sources

The following diagram illustrates DoS protection applied to the softswitch and to the Net-Net 9200.



Levels of DoS Protection

The multi-level Net-Net 9200 DoS protection consists of the following strategies:

- Fast path filtering/access control—Access control for signaling packets destined for the Net-Net 9200 host processors as well as media (RTP) packets. The Net-Net 9200 performs media filtering by using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted list and a deny list for each Net-Net 9200 realm (although the actual devices can be dynamically trusted or denied by the Net-Net 9200 based on configuration). You do not have to

provision every endpoint/device on the Net-Net 9200, but instead retain the default values.

- Host path protection—Includes flow classification, host path policing and unique signaling flow policing. Fast path filtering alone cannot protect the Net-Net 9200 host processors from being overwhelmed by a malicious attack from a trusted source. The host path and individual signaling flows must be policed to ensure that a rate-based attack will not overwhelm the Net-Net 9200's normal call processing; and subsequently not overwhelm systems beyond it.

The Net-Net 9200 must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent. At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Net-Net 9200 maintains two host paths, one for each class of traffic (trusted and untrusted), with different policing characteristics to ensure that fully trusted traffic always gets precedence.

- Host-based malicious source detection and isolation (dynamic deny list)—Malicious sources can be automatically detected in real-time and denied in the fast path to block them from reaching the host processor.

About the Process

DoS attacks are handled in the Net-Net 9200's host path. The Net-Net 9200 uses NAT table entries to filter out undesirable IP addresses; creating a deny list. After a packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager subsystem based on the sender's IP address. NAT table entries distinguish signaling packets coming in from different sources for policing purposes. The maximum number of policed calls that the Net-Net 9200 can support is 16K (on 32K CAM / IDT CAM).

The Traffic Manager has two pipes, trusted and untrusted, for the signaling path. Each signaling packet destined for the host CPUs traverses one of these two pipes.

Trusted Path

Packets from trusted devices travel through the trusted pipe in their own individual queues. In the Trusted path, each trusted device flow has its own individual queue (or pipe). The Net-Net 9200 can dynamically add device flows to the trusted list by promoting them from the Untrusted path based on behavior; or they can be statically provisioned.

Trusted traffic is put into its own queue and defined as a device flow based on the following:

- Source IP address
- Source UDP/TCP port number
- Destination IP address
- Destination UDP/TCP port (SIP interface to which it is sending)
- Realm it belongs to, which inherits the Ethernet interface and VLAN it came in on

For example, SIP packets coming from 10.1.2.3 with UDP port 1234 to the Net-Net 9200 SIP interface address 11.9.8.7 port 5060, on VLAN 3 of Ethernet interface 0:1, are in a separate Trusted queue and policed independently from SIP packets coming from 10.1.2.3 with UDP port 3456 to the same Net-Net 9200 address, port and interface.

Data in this flow is policed according to the configured parameters for the specific device flow, if statically provisioned. Alternatively, the realm to which endpoints belong have a default policing value that every device flow will use. The defaults configured in the realm mean each device flow gets its own queue using the policing values. As shown in the previous example, if both device flows are from the same realm and the realm is configured to have an average rate limit of 10K bytes per second (10KBps), each device flow will have its own 10KBps queue. They are not aggregated into a 10KBps queue.

The individual flow queues and policing lets the Net-Net 9200 provide each trusted device its own share of the signaling, separate the device's traffic from other trusted and untrusted traffic, and police its traffic so that it can't attack or overload the Net-Net 9200.

Address Resolution Protocol Flow

The Address Resolution Protocol (ARP) packets are given their own trusted flow with the bandwidth limitation of 8 Kbps. ARP packets are able to flow smoothly, even when a DoS attack is occurring.

Untrusted Path

Packets (fragmented and unfragmented) that are not part of the trusted or denied list travel through the untrusted pipe. In the untrusted path, traffic from each user/device goes into one of 2048 queues with other untrusted traffic. Packets from a single device flow always use the same queue of the 2048 untrusted queues, and 1/2048th of the untrusted population also uses that same queue. To prevent one untrusted endpoint from using all the pipe's bandwidth, the 2048 flows defined within the path are scheduled in a fair-access method. As soon as the Net-Net 9200 decides the device flow is legitimate, it will promote it to its own trusted queue.

All 2048 untrusted queues have dynamic sizing ability, which allows one untrusted queue to grow in size, as long as other untrusted queues are not being used proportionally as much. This dynamic queue sizing allows one queue to use more than average when it is available. For example, in the case where one device flow represents a PBX or some other larger volume device. If the overall amount of untrusted packets grows too large, the queue sizes rebalance, so that a flood attack or DoS attack does not create excessive delay for other untrusted devices.

In attack situations, the signaling processors detect the attack and dynamically demotes the device to be denied in the hardware by adding it to the deny ACL list. Even if the Net-Net 9200 does not detect an attack, the untrusted path gets serviced by the signaling processor in a fair access mechanism. An attack by an untrusted device will only impact 1/1000th of the overall population of untrusted devices, in the worst case. Even then there's a probability of users in the same 1/1000th percentile getting in and getting promoted to trusted.

IP Fragment Packet Flow

All fragmented packets are sent through their own 1024 untrusted flows in the Traffic Manager. The first ten bits (LSB) of the source address are used to determine which fragment-flow the packet belongs to. These 1024 fragment flows share untrusted bandwidth with already existing untrusted-flows. In total, there are 2049 untrusted flows: 1024-non-fragment flows, 1024 fragment flows, and 1 control flow.

Fragmented ICMP packets are qualified as ICMP packets rather than fragment packets. Fragmented and non-fragmented ICMP packets follow the trusted-ICMP-flow in the Traffic Manager, with a bandwidth limit of 8Kbs.

Static and Dynamic ACL Entry Limits

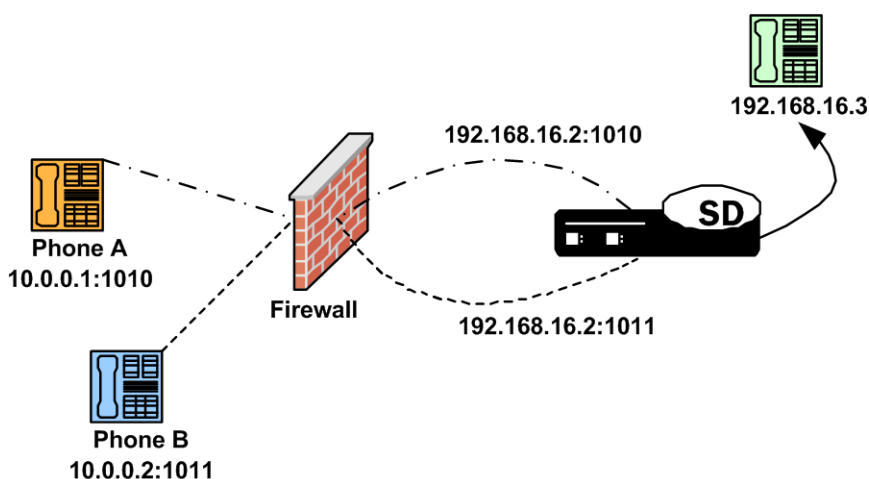
The Net-Net 9200 can simultaneously police a maximum of 16,000 trusted device flows, while at the same time denying an additional 16,000 attackers. If all 16,000 trusted entries and 16,000 denied entries are being used, the Net-Net 9200 can handle at most 32,000 simultaneous media flows (which is 16,000 simultaneous calls for normal voice calls). The usage is dynamic, so that when the Net-Net 9200 uses fewer trusted or denied entries, additional capacity for call media handling is freed. These limits let the Net-Net 9200 handle at least 32,000 simultaneous media flows, with the potential of up to 64,000. If list space becomes full and additional device flows need to be added, the oldest entries in the list are removed and the new device flows are added.

Dynamic Deny for HNT

Dynamic deny for HNT has been implemented on the Net-Net 9200 for cases when callers are behind a NAT or firewall. Without this feature, if one caller behind a NAT or firewall were denied, the Net-Net 9200 would also deny all other users behind the same NAT or firewall. This would be true even for endpoints behind the firewall that had not crossed threshold limits you set for their realm; all endpoints behind the firewall would go out of service. In the following diagram, both Phone A and Phone B would be denied because their IP addresses would be translated by the firewall to the same IP address (192.168.16.2).

However, dynamic deny for HNT allows the Net-Net 9200 to determine, based on the UDP/TCP port, which endpoints should be denied and which should be allowed. The Net-Net 9200 can determine that even though multiple endpoints originating behind a firewall appear with the same IP address, those addresses use different ports and are unique.

As shown in the diagram below, the ports from Phone A and Phone B remain unchanged. This way, if Phone A violates the thresholds you have configured, the Net-Net 9200 can block traffic from Phone A while still accepting traffic from Phone B.



Host and Media Path Protection Process

The Net-Net 9200 Network Processors (NPs) check the deny and permit lists for received packets, and classify them as trusted, untrusted or denied (discard). Only packets to signaling ports and dynamically signaled media ports are permitted. All other packets sent to Net-Net 9200 ports are filtered. Only packets from trusted and untrusted (unknown) sources are permitted; any packet from a denied source is

dropped by the NP hardware. The Traffic Manager manages bandwidth policing for trusted and untrusted traffic, as described earlier. Malicious traffic is detected in the host processor and the offending device is dynamically added to denied list, which enables early discard by the NP. Devices become trusted based on behavior detected by the Signaling Processor, and dynamically added to the trusted list. This process enables the proper classification by the NP hardware. All other traffic is untrusted (unknown).

Session Director Access Control

You can create static trusted/untrusted/deny lists with source IP addresses or IP address prefixes, UDP/TDP port number or ranges, and based on the appropriate signaling protocols. Furthermore, the Net-Net 9200 can dynamically promote and demote device flows based on the behavior, and thus dynamically creates trusted, untrusted, and denied list entries.

Access Control for Hosts

ACLs are only supported for SIP. The Net-Net 9200 loads ACLs so they are applied when signaling ports are loaded. The following rules apply to static NAT entries based on your configuration:

- If there are no ACLs applied to a realm that have the same configured trust level as that realm, the Net-Net 9200 adds a default NAT entry using the realm parameters.
- If you configure a realm with `none` as its trust level and you have configured ACLs, the Net-Net 9200 only applies the ACLs.
- If you set a trust level for the ACL that is lower than the one you set for the realm, the Net-Net 9200 will not add a separate NAT entry for the ACL.

ACLs provide access control based on destination addresses when you configure destination addresses as a way to filter traffic. You can set up a list of access control exceptions based on the source or the destination of the traffic.

For dynamic ACLs based on the promotion and demotion of endpoints, the rules of the matching ACL are applied.

Media Access Control

The media access control consists of media path protection and pinholes through the firewall. Only RTP and RTCP packets from ports dynamically negotiated through SIP signaling are allowed, which reduces the chance of RTP hijacking. Media access depends on both the destination and source RTP/RTCP UDP port numbers being correct, for both sides of the call.

Host Path Traffic Management

The host path traffic management consists of the dual host paths discussed earlier:

- Trusted path is for traffic classified by the Net-Net 9200 as trusted. You can initially define trusted traffic by ACLs, as well as by dynamically promoting it through a successful SIP registration, or a successful call establishment. You can configure specific policing parameters per ACL, as well as define default policing values for dynamically-classified flows. Traffic for each trusted device flow is limited from exceeding the configured values in hardware. Even an attack from a trusted, or spoofed trusted, device cannot impact the system.
- Untrusted path is the default for all unknown traffic that has not been statically provisioned otherwise. For example, traffic from unregistered endpoints. Pre-

configured bandwidth policing for all hosts in the untrusted path occurs on a per-queue and aggregate basis.

Traffic Promotion

Traffic is promoted from untrusted to trusted list when the following occurs:

- Successful SIP registration for SIP endpoints
- Successful session establishment for SIP calls

Malicious Source Blocking

Malicious source blocking consists of monitoring the following metrics for each source:

- SIP transaction rate (messages per second)
- SIP call rate (call attempts per second)
- Nonconformance/invalid signaling packets

Device flows that exceed the configured invalid signaling threshold, or the configured valid signaling threshold, within the configured time period are demoted, either from trusted to untrusted, or from untrusted to denied classification.

Blocking Actions

Blocking actions include the following:

- Dynamic deny entry added, which can be viewed through the ACLI.
- SNMP trap generated, identifying the malicious source

Dynamically added deny entries expire and are promoted back to untrusted after a configured default deny period time. You can also manually clear a dynamically added entry from the denied list using the ACLI.

Protecting Against Session Agent Overloads

You can prevent session agent (i.e., registrar) overloads with registrations by specifying the registrations per second that can be sent to a session agent.

ARP Flood Protection Enhancements

Enhancements have been made to the way the Net-Net SBC provides ARP flood protection. In prior releases, there is one queue for both ARP requests and responses, which the Net-Net SBC polices at a non-configurable limit (eight kilobytes per second). This method of ARP protection can cause problems during an ARP flood, however. For instance, gateway heartbeats the Net-Net SBC uses to verify (via ARP) reachability for default and secondary gateways could be throttled; the Net-Net SBC would then deem the router or the path to it unreachable, decrement the system's health score accordingly. Another example is when local routers send ARP requests for the Net-Net SBC's address are throttled in the queue; the Net-Net SBC never receives the request and so never responds, risking service outage.

The solution implemented to resolve this issue is to divide the ARP queue in two, resulting in one ARP queue for requests and a second for responses. This way, the gateway heartbeat is protected because ARP responses can no longer be flooded from beyond the local subnet. In addition, the Net-Net SBCs in HA nodes generate gateway heartbeats using their shared virtual MAC address for the virtual interface.

In addition, this solution implements a configurable ARP queue policing rate so that you are not committed to the eight kilobytes per second used as the default in prior releases. The previous default is not sufficient for some subnets, and higher settings resolve the issue with local routers sending ARP request to the Net-Net SBC that never reach it or receive a response.

As a security measure, in order to mitigate the effect of the ARP table reaching its capacity, configuring the media-manager option, **active-arp** is advised. Enabling this option causes all ARP entries to get refreshed every 20 minutes.

Dynamic Demotion of NAT Devices

As a method of protection from DoS attacks, the Net-Net SBC promotes and demotes devices based on the legitimacy of signaling messages received from them. When a device is communicating with the Net-Net SBC from behind a NAT, the Net-Net SBC can distinguish one device from another. You can enable the Net-Net SBC to block off an entire NAT device when a configured number of devices from behind a NAT are blocked off.

When this feature is enabled, the Net-Net SBC keeps track of the number of endpoints that have been black listed behind a NAT device. When the number of black listed endpoints behind a NAT device hits the configured threshold, the entire NAT device is black listed. A black listed NAT device remains black listed for the entire deny-period configured in the ACL or the realm.

This feature is enabled using the `nat-trust-threshold` parameter, which can be set either on an ACL configuration or on a per-realm basis. This functionality only applies to realms with a `trust-level` of `low` or `medium`.

ACLI Instructions and Examples

This section explains how to configure the dynamic demotion of NAT devices. You can enable this feature in both the access-control and realm-config configuration elements.

To set the dynamic demotion threshold for NAT devices for an access control element:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
```
3. Type **access-control** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# access-control
ACMEPACKET(access-control)#
```
4. **nat-trust-threshold**—Enter the number of individual devices behind the NAT that must be in the denied list before all the devices behind the NAT have their trust level set to denied. The default value is **0**, which means the dynamic demotion feature is disabled. The valid range is:
 - Minimum—0
 - Maximum—65535
5. Save and activate your configuration.

To set the dynamic demotion threshold for NAT devices for a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.

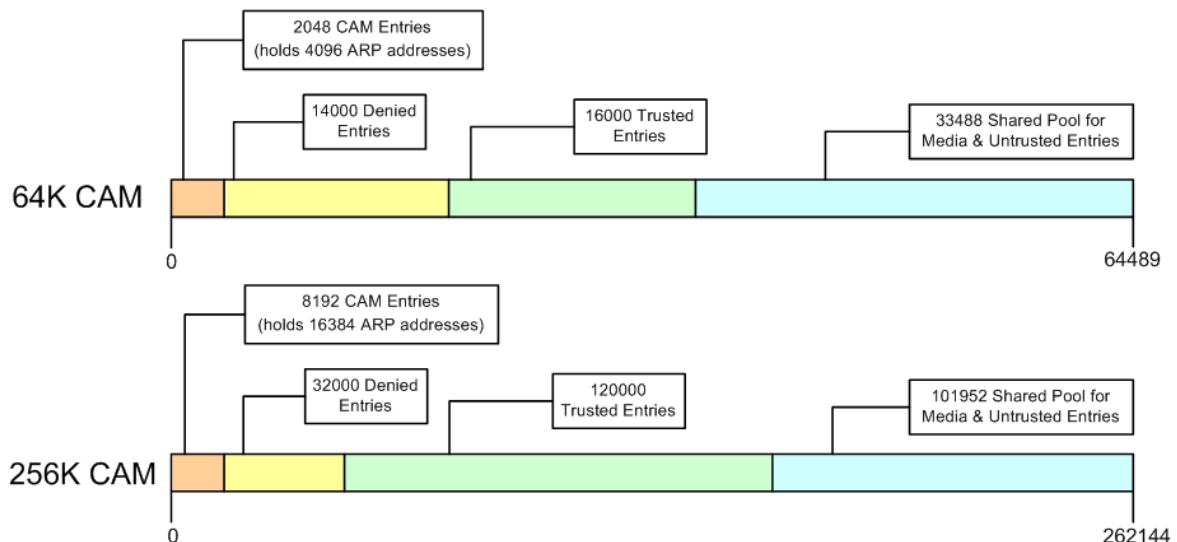

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **nat-trust-threshold**—Enter the number of individual devices behind the NAT that must be in the denied list before all the devices behind the NAT have their trust level set to denied. The default value is **0**, which means the dynamic demotion feature is disabled. The valid range is:
 - Minimum—0
 - Maximum—65535

CAM Size and Flow Apportioning

The Net-Net SBC is available with a 64K CAM and can be upgraded to a 256K CAM. The larger CAM increases the number of all entries available for each of the four flow classification queues as used in the DDoS (Net-SAFE) protection.

There is no configuration required to accommodate either of the two CAM models. However, depending on the installed CAM, different sized queues are used for each entry type. The following diagrams show the default queue sizes:



Entry Types

The CAM contains four pools of entry types. These are:

- ARP Entries
- Denied Entries
- Trusted Entries
- Media & Untrusted Entries

ARP Entries

The ARP address group is not configurable. It is fixed based on the CAM model in your system. The number of ARP addresses available is 2x the number of CAM entries allotted for this purpose.

Denied and Trusted Entries

The denied and trusted entry pools are each configurable. The trusted entries are configured as the max trusted allocation parameter and the denied entries are configured as the max deny allocation parameter. Denied entries are relinquished often enough so that free space is always available. After all trusted entries are used, the longest idle entry is reused by a new entry.

Media and Untrusted Entries

The media and untrusted entry pool is the remainder of the CAM's total entries after ARP, maximum denied, and maximum trusted entries have been configured (or left to default). Media and untrusted entries share the available entries in this group.

When all media and untrusted entries are in use, the Net-Net SBC must wait for one of these entries to be freed before a new one can be added.

The following table lists values to use for determining the Media & Untrusted pool:

	64K CAM	256K CAM
total available entries:	64489	262144
maximum trusted entry allocation:	0 - 16000	0 - 180000
maximum denied entry allocation:	0 - 14000	0 - 62000

Insert the above numbers in the following equation to determine the number of entries in the Media & Untrusted pool:

$$\text{Media \& Untrusted Pool} = \text{Total Available Entries} - (\text{Trusted Entries} + \text{Denied Entries})$$

Dynamic Reconfiguration

You can reconfigure maximum trusted and denied entry limits without a system reboot, but the new values will not take affect immediately if the existing CAM entries for either of these classes are filled. That is, the CAM entries will be gracefully removed until each queue does not exceed its allotted size.

When a system reconfiguration attempts to decrease the number of available trusted entries, the LRU (least recently used) mechanism is used to reduce the number of active trusted entries. LRU removes existing, over-the-limit trusted entries when any new trusted, denied or untrusted/media entry is created. Eventually, all extra trusted entries are removed from the CAM, freeing up space for the other types of entries.

When a system reconfiguration attempts to increase the number of available trusted entries, and CAM space is not available, all attempts to add new trusted entries fail until the space becomes available.

Because denied and untrusted & media entries have a relatively short timeout, they will most likely be relinquished quickly as free CAM space becomes available.

ACLI Configurations and Examples

You can change the Net-Net 9200's default CAM entry allocation. Refer to the previous table for valid ranges and default values.

To configure CAM entries:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **max-trusted-allocation**—Set the maximum number of trusted entries the CAM can hold.
5. **deny-allocation**—Set the maximum number of denied entries the CAM can hold.
6. Save and activate your configuration regarding dynamic reconfiguration constraints.

Configuring DoS Protection

This section explains how to configure the Net-Net 9200 for DoS protection.

Configuration Overview

Configuring Net-Net 9200 DoS protection includes masking source IP and port parameters to include more than one match and configuring guaranteed minimum bandwidth for trusted and untrusted signaling path. You can also configure signaling path policing parameters for individual source addresses. Policing parameters are defined as peak data rate (in bytes/sec), average data rate (in bytes/sec), and maximum burst size.

You can configure deny list rules based on the following:

- Ingress realm
- Source IP address
- Source port
- Transport protocol (TCP/UDP)
- Application protocol (SIP)

Changing the Default Net-Net 9200 Behavior

The Net-Net 9200 automatically creates permit untrusted ACLs that let all sources (address prefix of 0.0.0.0/0) reach each configured realm's signaling interfaces, regardless of the realm's address prefix. To deny sources or classify them as trusted, you create static or dynamic ACLs, and the global permit untrusted ACL to

specifically deny sources or classify them as trusted. Doing this creates a default permit-all policy with specific deny and permit ACLs based on the realm address prefix.

You can change that behavior by configuring static ACLs for realms with the same source prefix as the realm's address prefix; and with the trust level set to the same value as the realm. Doing this prevents the permit untrusted ACLs from being installed. You then have a default deny all ACL policy with specific static permit ACLs to allow packets into the system.

Example 1: Limiting Access to a Specific Address Prefix Range

The following example shows how to install a permit untrusted ACL of source 12.34.0.0/16 for each signalling interface/port of a realm called **access**. Only packets from within the source address prefix range 12.34.0.0/16, destined for the signaling interfaces/port of the realm named **access**, are allowed. The packets go into untrusted queues until they are dynamically demoted or promoted based on their behavior. All other packets are denied and dropped.

- Configure a realm called **access** and set the trust level to low and the address prefix to 12.34.0.0/16.
- Configure a static ACL with a source prefix of 12.34.0.0/16 with the trust level set to low for the realm named **access**.

Example 2: Classifying the Packets as Trusted

Building on Example 1, this example shows how to classify all packets from 12.34.0.0/16 to the realm signaling interfaces as trusted and place them in a trusted queue. All other packets from outside the prefix range destined to the realm's signaling interfaces are allowed and classified as untrusted; then promoted or demoted based on their behavior.

You do this by adding a global permit untrusted ACL (source 0.0.0.0) for each signaling interface/port of the **access** realm. You configure a static ACL with a source prefix 12.34.0.0/16 and set the trust level to high.

Adding this ACL causes the Net-Net 9200 to also add a permit trusted ACL with a source prefix of 12.34.0.0/16 for each signaling interface/port of the **access** realm. This ACL is added because the trust level of the ACL you just added is high and the realm's trust level is set to low. The trust levels must match to remove the global permit trusted ACL.

Example 3: Installing Only Static ACLs

This example shows you how to prevent the Net-Net 9200 from installing the global permit (0.0.0.0) untrusted ACL.

- Configure a realm with a trust level of **none**.
- Configure static ACLs for that realm with the same source address prefix as the realm's address prefix (e.g., 12.34.0.0/16), and set the trust level to any value.

The Net-Net system installs only the static ACLs you configure.

Configuring Access Control

To configure access control:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **access-control** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# access-control  
ACMEPACKET(access-control)#
```

4. **realm-id**—Enter the ID of the host's ingress realm.
5. **description**—Enter an optional description for this configuration element. If your entry contains any spaces, you must put the entire entry in double quotation marks.
6. **source-address**—Enter the source IP address and port number for the host in the following format:

```
<IP address>[/number of address bits][:<port>][/<port bits>]
```

For example:

```
10.0.0.1/24:5000/14
```

```
10.0.0.1/16
```

```
10.0.0.1/24:5000
```

```
10.0.0.1:5000
```

You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching.

7. **destination-address**—Ignored if you configure an application protocol in step 7. If not, enter the destination IP address and port for the destination in the following format:

```
<IP address>[/number of address bits][:<port>][/<port bits>]]
```

You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. For example, 192.168.0.10:5060.

8. **application-protocol**—Enter SIP as the application protocol type for this ACL entry. Valid values are
 - SIP | H.323 | MGCP | None

Note that if application-protocol is set to none, the destination-address and port will be used. Ensure that your destination-address is set to a non-default value (0.0.0.0.)

9. **transport-protocol**—Enter the IP protocol type. Available values are:

- ALL
- TCP
- UDP

10. **access**—Enter the access control type or trusted list based on the trust-level parameter configuration for this host. Available values are:

- **permit**—Puts the entry into the untrusted list; the entry is promoted or demoted according to the trust level set for this host. If you use this value, you are required to set the **trust-level** parameter.
- **deny**—Puts the entry in the deny list

11. **average-rate-limit**—Set the sustained rate in bytes per second for host path traffic from a trusted source within the realm. Available values are:

- minimum: 0 is the default. Policing is disabled.
 - maximum: 999999999
12. **trust-level**—Indicate the trust level for the host with the realm. This parameter is required when you set the **access** parameter to **permit**. Available values are:
- **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - **low**—Host can be promoted to the trusted list or demoted to the deny list.
 - **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - **high**—Host is always trusted and never demoted.
13. **invalid-signal-threshold**—Enter the acceptable number of invalid signaling message rate within the tolerance window. The value you enter here is only valid when the trust level is low or medium. Available values are:
- minimum: 0 is disabled (default)
 - maximum: 999999999
- If the number of invalid messages exceeds this value within the tolerance window, the host is demoted.
14. **maximum-signal-threshold**—Set the maximum number of signaling messages the host can send within the tolerance window. The value you enter here is only valid when the trust level is low or medium. Value range is:
- minimum: 0 is disabled (default)
 - maximum: 999999999
- If the number of messages received exceeds this value within the tolerance window, the host is demoted.
15. **untrusted-signal-threshold**—Set the maximum number of messages the untrusted host can send within the tolerance window. Use to configure different values for trusted and untrusted endpoints for valid signaling message parameters. Value range is:
- minimum: 0 is disabled (default)
 - maximum: 999999999
- The default value is 0.
16. **max-endpoints-per-nat**—Sets the maximum number of endpoints behind a NAT which can communicate with the Net-Net SBC. Anything above this number will be denied based on the access control scheme. Value range is:
- minimum: 0 is disabled (default)
 - maximum: 65535
17. **nat-invalid-message-threshold**—Sets the maximum number of invalid messages from endpoints behind a particular NAT. Exceeding this value will cause the Net-Net SBC to put the NAT entity in the deny list for the duration of the deny period. Value range is:
- minimum: 0 is disabled (default)
 - maximum: 65535
18. **deny-period**—Indicate the time period in seconds after which the entry for this host is removed from the deny list. Value range is:

- minimum: 0 seconds
- maximum: 999999999 seconds

The default value is 30 seconds.

The following example shows access control configured for an untrusted host in the external realm.

```
access-control
    realm-id                external
    source-address          192.168.200.215
    destination-address     192.168.10.2:5060
    application-protocol    SIP
    transport-protocol      UDP
    access                  permit
    average-rate-limit      4000
    trust-level             low
    invalid-signal-threshold 1
    maximum-signal-threshold 50
    untrusted-signal-threshold 4
    deny-period             30
```

The following example of how to configure an denied list entry:

```
access-control
    realm-id                external
    source-address          192.168.200.200
    destination-address     192.168.10.2:5060
    application-protocol    SIP
    transport-protocol      UDP
    access                  deny
    average-rate-limit      0
    trust-level             none
    invalid-signal-threshold 0
    maximum-signal-threshold 0
    untrusted-signal-threshold 0
    deny-period             0
```

Host Access Policing

You can configure the Net-Net 9200 to police the overall bandwidth of the host path.

To configure host access policing:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **max-signaling-bandwidth**—Set the maximum overall bandwidth available for the host path in bytes per second, which includes signaling messages from trusted and untrusted sources. It also includes any Telnet and FTP traffic on

media ports. Available values are as follows, and should be entered in increments (multiples) of 71,000:

- minimum: 71,000
 - maximum: 10,000,000 (default)
5. **max-untrusted-signaling**—Set the percentage of the maximum signaling bandwidth you want to make available for messages coming from untrusted sources. This bandwidth is only available when not being used by trusted sources. Available range of percentage values is:
 - minimum: 1
 - maximum: 100 (default)
 6. **min-untrusted-signaling**—Set the percentage of the minimum signaling bandwidth you want reserved for the untrusted sources. The rest of the bandwidth is available for trusted resources, but can also be used for untrusted sources (see max-untrusted-signaling). Range of values is:
 - minimum: 1
 - maximum: 100

The default value is 30.

7. **tolerance-window**—Set the size of the window used to measure host access limits. The value entered here is used to measure the invalid message rate and maximum message rate for the realm configuration. The default is 30 seconds.

The following example shows a host access policing configuration for the media manager (**media-manager-config**).

```
media-manager
    max-signaling-bandwidth      1000000
    max-untrusted-signaling      50
    min-untrusted-signaling      30
    tolerance-window             30
```

Configuring ARP Flood Protection

You do not need to configure the Net-Net SBC to enable the use of two separate ARP queues; that feature is enabled automatically.

If you want to configure the ARP queue policing rate, you can do so in the media manager configuration. Note that this feature is not RTC-supported, and you must reboot your Net-Net SBC in order for your configuration changes to take effect.

To set the ARP queue policing rate:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Enter **media-manager** and press <Enter>:.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **arp-msg-bandwidth**—Enter the rate at which you want the Net-Net SBC to police the ARP queue; the value you enter is the bandwidth limitation in bytes

per second. The minimum value is 2000, and the maximum is 200000; the default value is 32000.

5. Save your configuration.

Reboot your Net-Net SBC.

Access Control for a Realm

Each host within a realm can be policed based on average rate, peak rate, and maximum burst size of signaling messages. These parameters take effect only when the host is trusted. You can also set the trust level for the host within the realm. All untrusted hosts share the bandwidth defined for the media manager: maximum untrusted bandwidth and minimum untrusted bandwidth.

To configure access control for a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **addr-prefix**—Set the IP address prefix used to determine if an IP address is associated with the realm. This value is then associated with the ACLs you create to determine packet access. The default value is 0.0.0.0.
5. **average-rate-limit**—Set the sustained rate for host path traffic from a trusted source within the realm in bytes per second. Available values are:
 - minimum: 0 is disabled
 - maximum: 999999999
6. **access-control-trust-level**—Set the trust level for the host within the realm. The available values are:
 - **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - **low**—Host can be promoted to the trusted list or demoted to the deny list.
 - **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - **high**—Host is always trusted, and never demoted.
7. **invalid-signal-threshold**—Set the acceptable invalid signaling messages one host can send within the window of tolerance. If the Net-Net 9200 receives an invalid message more than the configured value within the tolerance window, the demotion is applied to the host. This parameter is only valid when the trust level is set to **low** or **medium**. Available values are:
 - minimum: 0 is disabled
 - maximum: 4999999999
8. **maximum-signal-threshold**—Set the maximum number of signaling messages one host can send within the window of tolerance. The host is demoted if the number of messages received by the Net-Net 9200 exceeds the

number set here. Valid only when the trust level is set to low or medium.
Available values are:

- minimum: zero (0) is disabled (default)
 - maximum: 999999999
9. **untrusted-signal-threshold**—Set the maximum number of messages the untrusted host can send within the tolerance window. Use to configure different values for trusted and untrusted endpoints for valid signaling message parameters. Value range is:
- minimum: 0 is disabled (default)
 - maximum: 999999999
10. **deny-period**—Set the length of time an entry remains on the deny list. The host is deleted from the deny list after this time period. Available values are:
- minimum: 0)
 - maximum: 999999999

The default value is 30.

The following example shows a host access policing configuration.

```

realm-config
  identifier                private
  addr-prefix               192.168.200.0/24
  network-interfaces

  mm-in-realm               private:0
  mm-in-network             disabled
  msm-release               enabled
  qos-enable                disabled
  max-bandwidth             disabled
  ext-policy-svr            0
  max-latency               0
  max-jitter                0
  max-packet-loss           0
  observ-window-size        0
  parent-realm
  dns-realm
  media-policy
  in-translationid
  out-translationid
  class-profile
  average-rate-limit        8000
  access-control-trust-level medium
  invalid-signal-threshold  1
  maximum-signal-threshold  50
  untrusted-signal-threshold 4
  deny-period               30
  symmetric-latching        disabled
  pai-strip                 disabled
  trunk-context

```

Configuring Overload Protection for Session Agents

To configure overload protection for session agents:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


```
ACMEPACKET(configure)# session-router
```
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```
4. **constraints**—Set the constraints parameter to **enabled** to set the sustained rate window constraint you configure in the next step.
5. **sustain-rate-window**—Enter a number to set the sustained window period (in milliseconds) that is used to measure the sustained rate. (Refer to the max-sustain-rate information).

The range of values is:

- minimum: 0 milliseconds (default)
- maximum: 999999999 milliseconds

6. **max-register-sustain-rate**—Enter a number to set the maximum number of registrations per second you want sent to the session agent. Also configurable per realm.

The range of values is:

- minimum: 0 seconds (default)
- maximum: 999999999 seconds

The following example shows a session agent configuration with overload protection.

```
session-agent
  hostname                192.168.1.10
  ip-address              192.168.1.10
  port                    5060
  state                   enabled
  app-protocol            SIP
  app-type
  transport-method        UDP
  realm-id                core
  description             registrar
  carriers
    carrier1
  allow-next-hop-lp       enabled
  constraints              disabled
  max-sessions             2000
  max-outbound-sessions   2000
  max-burst-rate           50
  max-sustain-rate        3000
  time-to-resume           60
  ttr-no-response         0
  in-service-period       30
  burst-rate-window       60
```

sustain-rate-window	3600
req-uri-carrier-mode	None
proxy-mode	Proxy
redirect-action	Recurse
loose-routing	enabled
response-map	
ping-method	
ping-interval	0
media-profiles	
in-translationid	
out-translationid	
trust-me	disabled
request-uri-headers	
stop-recurse	
local-response-map	
ping-to-user-part	
ping-from-user-part	
li-trust-me	disabled
in-manipulationid	
out-manipulationid	
p-asserted-id	
trunk-group	
max-register-sustain-rate	3600

DDoS Protection from Devices Behind a NAT

A DDoS attack could be crafted such that multiple devices from behind a single NAT could overwhelm the Net-Net SBC. The Net-Net SBC would not detect this as a DDoS attack because each endpoint would have the same source IP but multiple source ports. Because the Net-Net SBC allocates a different CAM entry for each source IP:Port combination, this attack will not be detected. This feature remedies such a possibility.

Restricting the Number of Endpoints behind a NAT

Each new source IP address and source IP port combination now counts as an endpoint for a particular NAT device. After the configured value of a single NAT's endpoints is reached, subsequent messages from behind that NAT are dropped and the NAT is demoted. This is set with the **max-endpoints-per-nat** parameter located in both the access-control and realm-config configuration elements.

Counting Invalid Messages from Endpoints behind a NAT

The Net-Net SBC also counts the number of invalid messages sent from endpoints behind the NAT. Once a threshold is reached, that NAT is demoted. Numerous conditions are counted as Errors/Invalid Messages from an endpoint. The aggregate of all messages from endpoints behind the NAT are counted against the NAT device, in addition to the existing count against the endpoint. This threshold is set with the **nat-invalid-message-threshold** parameter located in both the access-control and realm-config configuration elements.

As a unique case, the absence of a REGISTER message following a 401 response is counted as an invalid message from the end point. And if that endpoint is behind a NAT, this scenario will be counted as invalid message from that NAT device as well. You set a timeout period in which the REGISTER message must arrive at the Net-

Net SBC. This period is set with the **wait-time-for-invalid-register** parameter located in the realm config.

ACLI Instructions

To set the DDoS Protection from devices behind NATs:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. **max-endpoints-per-nat**—Set the maximum number of endpoints that can exist behind a NAT before demoting the NAT device. Valid values are 0-65535, with 0 disabling this feature. This parameter is also found in the access control configuration element.
5. **nat-invalid-message-threshold**—Set the maximum number of invalid messages that may originate behind a NAT before demoting the NAT device. Valid values are 0-65535, with 0 disabling this feature. This parameter is also found in the access control configuration element.
6. **wait-time-for-invalid-register**—Set the period (in seconds) that the Net-Net SBC counts before considering the absence of the REGISTER message as an invalid message.
7. Type **done** to save your work and continue.

SNMP Trap support

The following trap is sent by the Net-Net SBC whenever a NAT device is blacklisted due to the triggers listed in this feature. Reasons are not included in the trap, but are available from the syslogs.

```
apSysMgmtExpDOSTrap      NOTIFICATION-TYPE
OBJECTS                  { apSysMgmtDOSIpAddress,  apSysMgmtDOSRealmID ,
                           apSysMgmtDOSFromUri }
STATUS                    deprecated
DESCRIPTION
    "This trap is generated when an IP is placed on a deny list due
    to denial-of-service attempts, and provides the ip address that
    has been demoted, the realm-id of that IP, and (if available)
    the URI portion of the SIP From header of the message that
    caused the demotion."
::= { apSysMgmtDOSNotifications 2 }
```

Ensure that the **enable-snmp-monitor-traps** parameter in the system-config configuration element is enabled for the Net-Net SBC to send out this trap.

Syslog Support

Set the **syslog-on-demote-to-deny** parameter in the media-manager-config to enabled to generate syslog on endpoint demotion from untrusted to deny. NAT device demotion will also generate a unique syslog message with accompanying text explaining that it is the NAT device demotion event.

Debugging

The **show sip acl** command now includes counts of Blocked NAT devices.

```
ACMEPACKET# show sip acl
14:57:39-180 sip@s0.4.1
SIP Access Lists
```

	---- Recent ----			----- Lifetime -----		
	Active	High	Total	Total	PerMax	High
Total Entries	0	0	0	0	0	0
Trusted	0	0	0	0	0	0
Blocked	0	0	0	0	0	0
ACL Requests	-	-	0	0	0	0
Bad Messages	-	-	0	0	0	0
Promotions	-	-	0	0	0	0
Demotions	-	-	0	0	0	0
Blocked Nat Devices	0	0	0	0	0	0

Media Policing

Media policing controls the throughput of individual media flows in the Net-Net 9200, which in turn provides security and bandwidth management functionality. The media policing feature also lets you police static flows and RTCP flows.

The term *media policing*, as discussed in this section, refers to flows that go through the Net-Net 9200. Flows that are directed to the host application are not affected by media policing.

You can use media policing to protect against two potential security threats that can be directed toward your Net-Net 9200:

- **Media DoS**—Once media flows are established through the Net-Net 9200, network resources are open to RTP media flooding. You can eliminate the threat of a media DoS attack by constraining media flows to absolute bandwidth thresholds.
- **Bandwidth Piracy**—Bandwidth policing ensures that sessions consume no more bandwidth than what is signaled for.

Policing Methods

The Net-Net 9200 polices real-time traffic by using Constant Bit Rate (CBR) media policing. CBR policing is used when a media flow requires a static amount of bandwidth to be available during its lifetime. CBR policing best supports real-time applications that have tightly constrained delay variation. For example, voice and video streaming are prime candidates for CBR policing.

Session Media Flow Policing

Session media encompasses RTP and RTCP flows. In order to select policing constraints for these flows, the Net-Net 9200 watches for the codec specified in the SDP. When a match is made between the codec listed in an incoming session request and a configured media profile configuration element, the Net-Net 9200 applies that media profile's bandwidth policing constraint to the media flow about to start.

If multiple codecs are listed in the SDP message, the Net-Net 9200 will use the media profile with the most permissive media policing constraints for all of the flows associated with the session. If a codec in the SDP message is not found in any configured media profile, the Net-Net 9200 uses the media profile with the most permissive media policing constraints configured. If no media profiles are configured, there will be no session media flow policing.

If a mid-call change occurs, bandwidth policing is renegotiated.

Static Flow Policing

Static flows can also be policed in the same way as media flows are policed. A static flow configuration redirects flows entering the Net-Net 9200 on a media interface. The redirection is based on realm, source, destination, and protocol. When a flow matches the configured static flow criteria, besides being redirected toward a specified destination, its rate can also be controlled based on a static flow policing parameter found in the static flow configuration. Static flow policing operates obliviously to the data contained within the flow.

Fragmented IP Packet Policing

You can set the maximum amount of bandwidth (in the `max-untrusted-signaling` parameter) you want to use for untrusted packets. However, because untrusted and fragmented packets share the same amount of bandwidth for policing, any flood of untrusted packets can cause the Net-Net SBC to drop fragment packets.

To prevent fragment packet loss, you can set the **fragment-msg-bandwidth**. When it is set to any value other than 0 (which disables it), the Net-Net SBC:

- Provides for a separate policing queue for fragment packets (separate from that used for untrusted packets)
- Uses this new queue to prevent fragment packet loss when there is a flood from untrusted endpoints.

When you set up a queue for fragment packets, untrusted packets likewise have their own queue—meaning also that the `max-untrusted-signaling` and `min-untrusted-signaling` values are applied to the untrusted queue.

Enabling the fragmented message queue can also be used to shield untrusted traffic from a fragmentation attack.

Note: Enabling the fragmented message queue, i.e. changing the `fragment msg bandwidth` parameter from 0 to a non zero value requires a reboot to take effect. Changing the size of the queue once enabled does not require a reboot.

Configuration Notes

Review the following information before configuring your Net-Net 9200 to perform media policing.

Session Media Flow Policing

Session media flow policing applies to both RTP and RTCP flows. Setting either of the parameters listed below to 0 disables media policing, letting RTP or RTCP flows pass through the Net-Net 9200 unrestricted.

- RTP Policing
 - Set in the **media-profile** configuration's **average-rate-limit** parameter to police RTP traffic with the CBR policing method.

- **average-rate-limit**—Establishes the maximum bandwidth for a flow in bytes per second.
- RTCP Policing
 - Set in the **media-manager-config** configuration element's **rtcp-rate-limit** parameter to police RTCP traffic with the CBR policing method.
 - **rtcp-rate-limit**—Establishes the maximum speed for an RTCP flow in bytes per second.

Static Flow Policing

Static flow policing is configured with one parameter found in the **static-flow** configuration element. To configure CBR, you have to set the **average-rate-limit** parameter to a non-zero value. Setting the parameter listed below to 0 disables static flow policing, effectively letting the flow pass through the Net-Net 9200 unrestricted.

In a CBR configuration, the **average-rate-limit** parameter determines the maximum bandwidth available to the flow.

- **average-rate-limit**—Establishes the maximum bandwidth for a static flow in bytes per second.

Note: Static flow policing is not necessarily tied to any type of media traffic, it can affect flows of any traffic type.

Configuring Media Policing for RTP Flows

This section show you how to configure media policing in the media profile configuration.

ACLI Instructions and Examples

In the following example, you will configure media policing for the G723 media profile.

To configure media policing for RTP flows:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# session-router
```
3. Type **media-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(session-router)# media-profile
```
4. Select an existing media profile to which you will add policing constraints.


```
ACMEPACKET(media-profile)# select
<name>:
1: audio    4=G723      RTP/AVP 16 0 0 0

selection:1
ACMEPACKET(media-profile)#
```

From this point, you can configure media policing parameters.
5. **average-rate-limit**—Enter the maximum rate in bytes per second for any flows that this **media-profile** polices. Valid values for this parameter:

- Minimum: 0 (media policing disabled)
- Maximum: 125000000
- Default: 0

Average rate limit values for common codecs:

- PCMU—80000 Bps
- G729—26000 Bps

The following example shows a **media-profile** configuration element configured for media policing for codec G.723..

```
media-profile
  name                G723
  media-type          audio
  payload-type        4
  transport            RTP/AVP
  req-bandwidth        16
  frames-per-packet    0
  parameters
    average-rate-limit 15000
```

Configuring Media Policing for RTCP Flows

This section show you how to configure media policing for RTCP flows.

ACLI Instructions and Examples

To configure media policing for RTCP flows:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the **media-manager** path.


```
ACMEPACKET(configure)# media-manager
```
3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **rtcp-rate-limit**—Enter the RTCP policing constraint in bytes per second. Valid values for this parameter:
 - Minimum: 0
 - Maximum: 125000000
 - Default: 0

Configuring Media Policing for Static Flows

This section shows you how to configure media policing for static flows.

ACLI Instructions and Examples

To configure media policing for static flows:

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

```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press <Enter> to access the **media-manager** path.

```
ACMEPACKET(configure)# media-manager
```

3. Type **static-flow** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# static-flow
```

```
ACMEPACKET(static-flow)#
```

4. Select an existing static flow to which you will add policing constraints.

```
ACMEPACKET(static-flow)# select
```

```
<in-dest-ip>:
```

```
1: dest 0.0.0.0; src 192.168.2.1/24; static-flow-in-realm; UDP
```

```
selection:1
```

5. **average-rate-limit**—Enter the maximum rate in bytes per second for any flows that this **static-flow** polices. Valid values for this parameter:

- Minimum: 0 (media policing disabled)
- Maximum: 125000000
- Default: 0

The following example shows a **static-flow** configuration element configured for media policing.

```
static-flow
```

in-realm-id	access
in-source	192.168.2.1/24
in-destination	0.0.0.0
out-realm-id	core
out-source	192.168.128.1/24
out-destination	0.0.0.0
protocol	UDP
average-rate-limit	15000

Configuring Media Policing for Fragmented Packets

This section shows you how to configure policing for fragmented packets.

ACLI Instructions and Examples

To enable fragmented IP packet policing:

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

```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET(configure)# media-manager
```

3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# media-manager
```

```
ACMEPACKET(media-manager-config)#
```

4. **fragment-msg-bandwidth**—Enter the amount of bandwidth to use for the fragment packet queue. If you leave this parameter set to 0, then the Net-Net SBC will use the same queue for and share bandwidth between untrusted packets and fragment packets. The default value is zero (0). Valid policed values range between 2,000-10,000,000 Bytes/sec.
5. Save and activate your configuration.

RTP Payload Type Mapping

The Net-Net 9200 maintains a default list of RTP payload types mapped to textual encoding names as defined in RFC 3551.

The following table defines the preconfigured payload type for standard encodings.

Payload Type	Encoding Name	Audio (A) / Video (V)	Clock Rate
0	PCMU	A	8000
4	G723	A	8000
8	PCMA	A	8000
9	G722	A	8000
15	G728	A	8000
18	G729	A	8000

If you configure any payload type to encoding name mappings, the default mappings will be ignored. You must then manually enter all payload type mappings you use in the **media-profile** configuration element.

TCP Synchronize Attack Prevention

This section explains how the Net-Net SBC protects itself from a Transmission Control Protocol (TCP) synchronize (SYN) packet flooding attack sourced from a remote hostile entity.

SIP signaling can be configured on the Net-Net SBC to be TCP protocol-based. In this configuration, the Net-Net SBC can be a target of a TCP SYN attack. The Net-Net SBC is able to service new call requests throughout the duration of an attack

About SYN

SYN is used by TCP when initiating a new connection to synchronize the sequence numbers on two connecting computers. The SYN is acknowledged by a SYN-ACK by the responding computer. After the SYN-ACK, the client finishes establishing the connection by responding with an ACK message. The connection between the client and the server is then open, and the service-specific data can be exchanged between the client and the server.

A SYN flood is a series of SYN packets from forged IP addresses. The IP addresses are chosen randomly and do not provide any hint of the attacker's location. The SYN flood keeps the server's SYN queue full. Normally this would force the server to drop connections. A server that uses SYN cookies, however, will continue operating normally. The biggest effect of the SYN flood is to disable large windows.

Server Vulnerability

Vulnerability to attack occurs when the server has sent a SYN-ACK back to client, but has not yet received the ACK message; which is considered a half-open connection. The server has a data structure describing all pending connections built

in its system memory. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections.

The attacking system sends SYN messages to the server that appear to be legitimate, but in fact reference a client that is unable to respond to the SYN-ACK messages. The final ACK message is never sent to the server.

The half-open connections data structure on the server fills and no new incoming connections are accepted until the table is emptied out. Typically there is a timeout associated with a pending connection (the half-open connections will eventually expire and the server will recover). But the attacking system can continue sending IP-spoofed packets requesting new connections faster than the server can expire the pending connections. The server has difficulty in accepting any new incoming network connections.

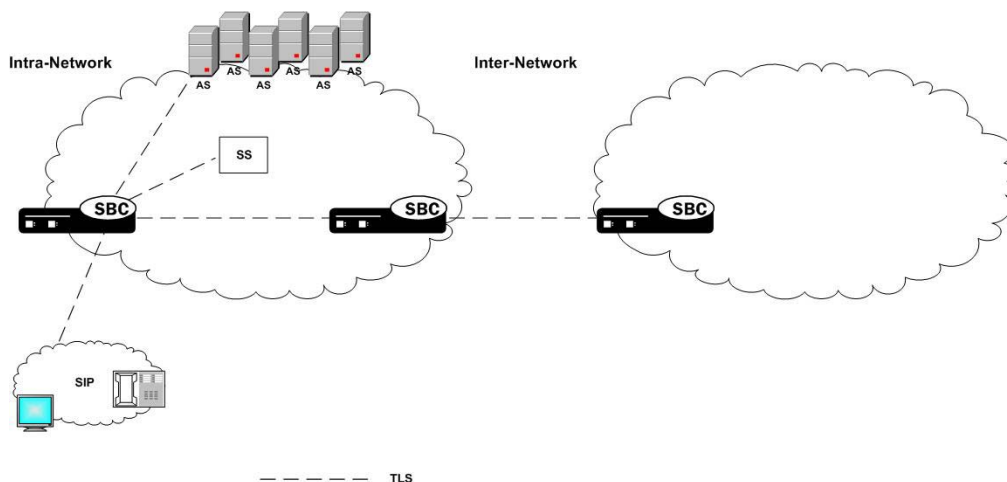
Configuring TCP SYN Attack Prevention

No configuration is necessary to enable TCP SYN attack prevention. Internal TCP protocol changes were made to provide protection.

Transport Layer Security

The Net-Net SBC provides support for Transport Layer Security (TLS) for SIP, which can be used to protect user and network privacy by providing authentication and guaranteeing the integrity for communications between the Net-Net SBC and the following:

- Another device in your network infrastructure (intra-network)
- Another Net-Net SBC when you are using a peering application (inter-network) for interior network signaling security
- An endpoint for authentication before allowing SIP messaging to take place



The Net-Net SBC and TLS

The Net-Net SBC's TLS functionality depends on the presence of a the Signaling Security Module (SSM) for hardware acceleration of encryption and decryption and random media generation. The SSM is a plug-on module that can be added to your Net-Net 9200's SPU given the installation of the necessary bootloader and minimum hardware revision levels.

With the requisite hardware revision levels, the plug-on unit can be added to an SPU in the field by qualified personnel. This provision makes upgrades fast, forgoing the need for you to return your Net-Net SBC to Acme Packet manufacturing for hardware upgrade. When your Net-Net SBC is upgraded with the SSM card that supports TLS, a new CLEI code will be added to your chassis; the code will also appear on the SSM card (also referred to as the plug-on unit) and if you look at new Net-Net SBCs outfitted with the SSM card, they will have the code labels already affixed in all required locations.

Unlike SPU1 which requires optional SSM2 hardware for encrypting SIP-TLS signaling, the SPU2 includes (4) built-in security co-processors for functions performed by SSM2.

TLS support will not behave in the manner described here if you do not have the SSM component installed on your Net-Net SBC, because it is the presence of this hardware that enables the TLS software support.

The accelerator card performs:

- RSA
- Diffie-Hellman
- 3DES
- AES256
- Random number generation

TLS Features

The Net-Net SBC supports the following TLS features:

- TLSv1/SSLv3
- RFC 3261 specific SIPs and TLS support in SIP
- Importing X509v3 certificates in PKCS7 / PKCS12 / X509v3 PEM / Base64 format
- Generating a private key and a certificate request in PKCS-10 PEM/Base64 format
- Displaying imported certificates in text format
- Configuration verification, including verification that all dependencies are resolved
- Connection reuse draft (draft-ietf-sip-connect-reuse-03.txt)
- HA for TLS—When the active card fails, the standby has the same TLS-related configuration.
 - Existing active calls are not affected by a failover—Enduser experiences no interruption or disturbance in service. SIP signaling messages sent over the connection following failover do not impact the active call.
 - New calls are made through the newly active card.
- Real-time reconfiguration (RTC)

The Net-Net SBC does not support certificate revocation listing handling.

Domestic and International Versions

There are two versions of the Net-Net OS that support TLS: a U.S. version and an international version. Two versions exist because of the laws governing the strength

of algorithms that can be shipped domestically and internationally. If you require further information, consult with your Acme Packet sales representative directly.

Supported Encryption

The Net-Net SBC provides support for TLSv1 and SSLv3 encryption.

TLSv1 Ciphers

The Net-Net SBC supports the TLS v1 cipher suites listed in this section.

For encryption, the Net-Net SBC supports: AES-128, AES-256, 3DES and ARC4 (40 and 128 bit) algorithms. It also supports:

- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLSv1
- SSLv3
- ALL [default]
- NONE

Mapping SSL3 to TLSv1 Ciphers

The following table shows the mapping of SSL3 ciphers to TLSv1 ciphers:

SSL3	TLSv1
SSL_RSA_WITH_NULL_MD5	TLS_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA	TLS_RSA_WITH_NULL_SHA
SSL_RSA_WITH_RC4_128_MD5	TLS_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA	TLS_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

Note: The Net-Net SBC supports **TLS_RSA_WITH_NULL_MD5** and **TLS_RSA_WITH_NULL_SHA** although neither does any encryption. These ciphers might be used for debugging purposes, and should not be deployed.

Signaling Support

The Net-Net SBC's TLS functionality supports SIP and SIPS. In addition, the Net-Net SBC can accommodate a mixture of TLS and non-TLS sessions within a realm because a request for TLS is controlled by the endpoint (TLS UA).

DoS Protection

The Net-Net SBC provides DoS protection from:

- Too many simultaneous TLS connections being requested by a single IP address.

The Net-Net SBC limits the number of TLS connections from a single IP address; you can set a maximum simultaneous number of TCP/TLS connections a SIP interface will allow from a single IP address.

- Too many simultaneous TLS connection requests by limiting the maximum number of connections for a SIP interface.

In other words, the maximum simultaneous TCP/TLS connections a SIP interface will allow in aggregate from all IP addresses served by that signaling interface.

- Endpoints establishing TCP/TLS connections that never send any messages (application layer messages; once the TLS handshake completes).

This protection is triggered by inactivity, measured by lack of any message from a peer. The value specified for this timer is in seconds.

- Endpoints requesting an initial registration that never send messages thereafter.

Note: It is expected that whenever an endpoint establishes a TCP/TLS connection, it will keep the connection active by sending additional messages or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level "ERROR" is generated.

- Malformed packets by counting and limiting the maximum number of malformed packets.

Whenever the Net-Net SBC receives an invalid TLS message, it increments the internal invalid signalling threshold counter. When that counter reaches the configured value, the Net-Net SBC denies the endpoints for the configured deny period. This also requires configuration of tolerance window in media manager.

Endpoint Authentication

The Net-Net SBC does not operate as a CA. Instead, the Net-Net SBC's TLS implementation assumes that you are using one of the standard CAs for generating certificates:

- Verisign
- Entrust
- Thawte
- free Linux-based CA (for example, openssl)

The Net-Net SBC can generate a certificate request in PKCS10 format and to export it. It can also import CA certificates and a Net-Net SBC certificate in the PKCS7/X509 PEM format.

The Net-Net SBC generates the key pair for the certificate request internally. The private key is stored as a part of the configuration in 3DES encrypted form (with an internal generated password) and the public key is returned to the user along with other information as a part of PKCS10 certificate request.

The Net-Net SBC supports the option of importing CA certificates and marking them as trusted. However, the Net-Net SBC only authenticates client certificates that are issued by the CAs belonging to its trusted list. If you install only a specific

vendor's CA certificate on the Net-Net SBC, it authenticates that vendor's endpoints. Whether the certificate is an individual device certificate or a site-to-site certificate does not matter because the Net-Net SBC authenticates the signature/public key of the certificate.

Key Usage Control

You can configure the role of a certificate by setting key usage extensions and extended key usage extensions as described in RFC 3280. Both of these are configured in the certificate record configuration.

Key Usage List

This section defines the values you can use (as a list) in the **key-usage-list** parameter. You can configure the parameter with more than one of the possible values.

Value	Description
digitalSignature (default with keyEncipherment)	Used when the subject public key is used with a digital signature mechanism to support security services other than non-repudiation, certificate signing, or revocation information signing. Digital signature mechanisms are often used for entity authentication and data origin authentication with integrity.
nonRepudiation	Used when the subject public key is used to verify digital signatures that provide a non-repudiation service protecting against the signing entity falsely denying some action, excluding certificate or CRL signing.
keyEncipherment (default with digitalSignature)	Used when the subject public key is used for key transport. (For example, when an RSA key is to be used for key management.)
dataEncipherment	Used when the subject public key is used for enciphering user data other than cryptographic keys.
keyAgreement	Used when the subject public key is used for key agreement. (For example, when a Diffie-Hellman key is to be used for a management key.)
encipherOnly	The keyAgreement type must also be set when this option is chosen. Used with the subject public key is used only for enciphering data while performing key agreement.
decipherOnly	The keyAgreement type must also be set when this option is chosen. Used with the subject public key is used only for deciphering data while performing key agreement.

Extended Key Usage List

This section defines the values you can use in the **extended-key-usage-list** parameter.

Value	Description
serverAuth (default)	Used when the certificate is used for TLS server authentication. In Net-Net SBC access-side deployments, the Net-Net SBC typically acts as a TLS server accepting TLS connections. You might use this setting while generating the end-entity-cert.
clientAuth	Used when the certificate is used for TLS client authentication. In Net-Net SBC core-side deployments, the Net-Net SBC typically acts as a TLS client initiating TLS connections. You might use this setting while generating the end-entity-cert.

Configuring TLS

This section explains how to configure your Net-Net SBC for TLS support.

Process Overview

In summary, you need to take the following steps to enable your Net-Net SBC for TLS.

1. Make sure that your Net-Net SBC has the appropriate hardware installed and that you have obtained an enabled the licenses related to TLS support.
2. Configure certificates.
3. Configure the specific parameters related to TLS.

Configuring Certificates

Configuring certificates is a three-step process:

1. Create a certificate record configuration on the Net-Net SBC
2. Generate a certificate request by the Net-Net SBC and save the configuration
3. Import the certificate record into the Net-Net SBC and save the configuration

Configuring the Certificate Record

The certificate record configuration represents either the end-entity or the CA certificate on the Net-Net SBC.

If it is used to present an end-entity certificate, a private key should be associated with this certificate record configuration using the ACLI **security certificate request** command.

No private key should be associated with the certificate record configuration if it was issued to hold a CA certificate.

A certificate can be imported to a certificate record configuration using the ACLI **security certificate import** command.

The following is sample of the certificate record configuration parameters as seen in the ACLI.

```
certificate-record
name          certificate record name
country       country name
state         state name
```

locality	locality name
organization	organization name
unit	organization unit
common-name	common name
key-size	key size
alternate-name	alternate name
trusted	certificate-record trusted or not

Note: There is no need to create a certificate record when importing a CA certificate or certificate in pkcs12 format.

To enter a certificate record using the ACLI configuration menu:

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


```
ACMEPACKET# configure terminal
```
2. Type **security** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# security
```
3. Type **certificate-record** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(security)# certificate-record
ACMEPACKET(certificate-record)#
```
4. **name**—Enter the name of the certificate record. This is a key field, and you must enter a value for it. For example, **acmepacket**.
5. **country**—Enter the name of the country. The default is **US**.
6. **state**—Enter the name of the state of for the country. The default is **MA**.
7. **locality**—Enter the name of the locality for the state. The default is **Burlington**.
8. **organization**—Enter the name of the organization holding the certificate. The default is **Engineering**.
9. **unit**—Enter the name of the unit for the holding the certificate within the organization.
10. **common-name**—Enter the common name for the certificate record.
11. **key-size**—Enter the size of the key for the certificate. Use the default of 1024, or change it to one of the other supported values: 512, 2048, or 4096.
12. **alternate-name**—Enter the alternate name of the certificate holder.
13. **key-usage-list**—Enter the usage extensions you want to use with this certificate record. This parameter can be configured with multiple values, and it defaults to the combination of **digitalSignature** and **keyEncipherment**. For a list of possible values and their descriptions, refer to the [Key Usage List \(723\)](#) section.
14. **extended-key-usage-list**—Enter the extended key usage extensions you want to use with this certificate record. The default is **serverAuth**. For a list of possible values and their descriptions, refer to the [Extended Key Usage List \(724\)](#) section.
15. Type **done** to save this configuration.
16. Save and activate the configuration from the superuser prompt.

Generating a Certificate Request

Using the ACLI **security certificate request** command allows you to generate a private key and a certificate request in PKCS10 PEM format. You can only perform this step once you have configured a certificate record.

The Net-Net SBC stores the private key that is generated in the certificate record configuration in 3DES encrypted form with an internally generated password. The PKCS10 request is displayed on the screen in PEM (Base64) form.

You use this command for certificate record configurations that hold end-entity certificates. If you have configured the certificate record to hold a CA certificate, then you do not need to generate a certificate request because the CA publishes its certificate in the public domain. You import a CA certificate by using the ACLI **security certificate request import** command.

This command sends information to the CA to generate the certificate, but you cannot have Internet connectivity from the Net-Net SBC to the Internet. You can reach the Internet through a browser such as Internet Explorer if it is available, or you can save the certificate request to a disk and then submit it to the CA.

To run the applicable command, you must use the value you entered in the name parameter of the certificate record configuration. You run the command from main Superuser mode command line, and then save and activate the configuration:

```
ACMEPACKET# security certificate request acmepacket
Generating Certificate Signing Request. This can take several
minutes....

-----BEGIN CERTIFICATE REQUEST-----
MIIB2jCCAUMCAQAwYTELMAkGA1UEBhMCdXMxCzAJBgNVBAGTAK1BMRMwEQYDVQQH
EwpCdXJsaw5ndG9uMRQwEgYDVQQKEwtFbmdpbmVlcmluZzEMMAoGA1UECzMdYWJj
MQwwCgYDVQQDEWNhYmMwZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALOMLHo8
/qI0ddIDVuqot0Y72l/BfH8loLRKmhZQ4e7sS+zZHzbG8phzmzhf0SECnZiA2bEo
f+Nti7e7Uof4LLwiYl9fvhURfzhENOKThAPKPiJCzBBglTITHTYa l00Cq2fj5A8B
ZcuAHj7Vp5wP2zpz6EUTFpqTDMlVdwJGJrElAgMBAAGgOTAMBGNVHRExBRMDZGVm
MCKGA1UdDzEiEyBkawdpdGFsU2lnbmF0dXJlLlGtleUVuY2lwaGVybWVudDANBgkq
hkiG9w0BAQUFAA0BgQAteL4ZSLI8gggMzodbYwGUHUGqTGeDzQDhJV5fKUXWeMFz
JsTmWn5Gy/kR4+Nq274G14fnk00fTAFmtgQ5aL3gM43TqaPOTZjJ6qgwURKHoBPI
7hkovkgAxHge7wClghiAp/ELdl7tQ515k04BMD5f/fxG7nNiU8iEg7P000IBgg==
-----END CERTIFICATE REQUEST-----

WARNING: Configuration changed, run "save-config" command.
ACMEPACKET# save config
copying file /code/config/dataDoc.gz -> /code/config/dataDoc_3.gz
copying file /code/config/tmp/editing/dataDoc.gz ->
/code/config/dataDoc.gz
Save complete
ACMEPACKET# activate config

activate complete
```

Importing a Certificate Using the ACLI

For an end-entity certificate, once a certificate is generated using the ACLI **security certificate request** command, that request should be submitted to a CA for generation of a certificate in PKCS7 or X509v3 format. When the certificate has been generated, it can be imported into the Net-Net SBC using the **security certificate import** command.

The syntax is:

```
ACMEPACKET # security certificate import [try-all | pkcs7 | pkcs12 |
x509] [certificate-record file-name]
```

To import a certificate:

1. When you use the **security certificate import** command, you can specify whether you want to use PKCS7, PKCS12, X509v3 format, or try all. In the command line, you enter the command, the format specification, and the name of the certificate record. Next, the Net-Net SBC will prompt you to enter the certificate in PEM format. Paste the certificate in the ACLI at this point. For example:

```
ACMEPACKET# security certificate import try-all acme
```

The following will appear:

Please enter the certificate in the PEM format.

Terminate the certificate with ";" to exit.....

```
-----BEGIN CERTIFICATE-----
```

```
MIIDHzCCAoigAwIBAgIIAhMCUACEAHewDQYJKoZIhvcNAQEFBQAwDELMAkGA1UE
BhMCVVMxEzARBgNVBAgTCkNhbgGmb3JuawExETAPBgNVBACTCFNhbiBkb3NlMQ4w
DAYDVQQKEwVzaXBpdDEpMCCGA1UECzMgU2lwaXQgVGZvdCBDZXJ0awZpY2F0ZSBB
dXRob3JpdHkwHhcNMDUwNDEzMjEzNzQzWhcnMDgwNDEyMjEzNzQzWjBUMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCTUEwEzARBgNVBACTCkUjcmxpbnmd0b24xZDASBgNV
BAoTC0Vuz2luZWVyaW5nMQ0wCwYDVQQDEwRhY21lMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCXjIe0yFKAUB3rKkKK/+59LT+rLGuw7Lgc1V6+hFTSr0co+ZsQ
bHFUWAA15qXUUBTLJG13QN5VfG96f7gGAbWayf0S9Uymold3JPCUD0Ggb2E7m8iu
vtq7gwjSeKNXAw/y7ywy/c04FmUD2U0pZX0CNIR3Mns50AxQmq0bNYDhawIDAQAB
o4HdMIHaMBEGA1UdEQKMAiCBnBrdw1hcjAJBgNVHRMEAjaAMB0GA1UdDgQWBbTG
tpodxa6Kmmn04L3Kg62t8BZJHTCBmgYDVR0jBIGSMIGPgBRrRhcu6pR2JYUBhNU
2qHjVBShtqF0pHIwcDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbgGmb3JuawEx
ETAPBgNVBACTCFNhbiBkb3NlMQ4wDAYDVQQKEwVzaXBpdDEpMCCGA1UECzMgU2lwa
aXQgVGZvdCBDZXJ0awZpY2F0ZSBBdXRob3JpdHmCAQAwDQYJKoZIhvcNAQEFBQAD
gYEAbs8nUCi+cA2hC/LM49Sith8Qmpl81KONApsoC4Em24L+DZwz3uInowbjJJ
QhefcUfteNYkbuMH7LAK0hndPvW+St4rQGVK6LJhZj7/yeLXmYWIPIUY3Ux40GVrd
2UgV/B2S0qH9Nf+FQ+mNZ0LL7EuF4IxSz9/69LuYLXqKSG4=
```

```
-----END CERTIFICATE-----;
```

Certificate imported successfully....

WARNING: Configuration changed, run "save-config" command.

2. Save your configuration.

```
ACMEPACKET# save config
```

```
copying file /code/config/dataDoc.gz -> /code/config/dataDoc_3.gz
```

```
copying file /code/config/tmp/editing/dataDoc.gz ->
/code/config/dataDoc.gz
```

Save complete

3. Synchronize and activate your configurations.

```
ACMEPACKET# activate config
```

activate complete

Importing a Certificate Using FTP

You can also put the certificate file in the directory `/code/certs` and then executing the **security certificate import** command or by pasting the certificate in the PEM/Base64 format into the ACLI. If you paste the certificate, you might have to copy and paste it a portion at a time rather than pasting in the whole thing at once.

A password is required for certificates in pkcs12 format.

To import the certificate using FTP:

1. FTP the certificate file on to the Net-Net SBC (directory `/code/certs`), let us say the name of the certificate file is `cert.pem`.
2. Once the certificate is successfully transferred to the Net-Net SBC, run the **import-certificate** command.

The syntax is:

```
ACMEPACKET# security certificate import [try-all | pkcs7 | pkcs12 | x509] <certificate-record-name> <filename> <password>
```

Using the command will look like this when you have used FTP.

```
ACMEPACKET# import-certificate try-all acme cert.pem
```

```
Certificate imported successfully....
```

```
WARNING: Configuration changed, run "save-config" command.
```

3. Save your configuration.

```
ACMEPACKET# save config
```

```
copying file /code/config/dataDoc.gz -> /code/config/dataDoc_3.gz
```

```
copying file /code/config/tmp/editing/dataDoc.gz -> /code/config/dataDoc.gz
```

```
Save complete
```

4. Synchronize and activate your configurations.

```
ACMEPACKET# activate config
```

```
activate complete
```

```
ACMEPACKET#
```

Verifying a Certificate Record

Verifying a certificate in the certificate record is important due to the nature of the certificate. Various factors affect the validity of the certificate. It has certain lifetime, it can be revoked, and the password that used to protect the certificate could change.

```
ACMEPACKET# security certificate verify t6
```

```
verifying certificate-record (t6)
```

```
no chaining information, assume it is valid
```

```
no crl, assume it is valid
```

```
in sync with configuration password
```

TLS Version

This section shows you how to configure the TLS version. Since you can control the TLS version with a parameter, the TLSv1 and SSLv3 values have been deprecated from the **cipher-list** parameter. The cipher list now supports the following values:

- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_RSA_WITH_DES_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- ALL (default)
- NONE

Configuring a TLS Profile

The TLS profile configuration has been added to the security section of the ACLI's configure terminal menu. This configuration holds the information required to run SIP over TLS.

In the ACLI menu for this configuration, the parameters appear as follows:

```

tls-profile
  name                tls profile name
  end-entity-certificate  end entity certificate for the TLS connection
  trusted-ca-certificates list of trusted certificate records
  cipher-list          list of ciphers
  verify-depth          maximum length of the certificate chain
  mutual-authenticate  mutually authenticate

```

To configure a TLS profile:

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


```
ACMEPACKET# configure terminal
```
2. Type **security** and press <Enter> to access the **session-router** path.


```
ACMEPACKET(configure)# security
```
3. Type **tls-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)#
```
4. **name**—Enter the name of the TLS profile. This parameter is required; you cannot leave it empty.
5. **end-entity-certificate**—Enter the name of the entity certification record.
6. **trusted-ca-certificates**—Enter the names of the trusted CA certificate records.

7. **cipher-list**—Either use the default **ALL**, or enter a list of ciphers you want to support. Refer to the [Supported Encryption \(721\)](#) section of this chapter for more details.
8. **verify-depth**—Specify the maximum depth of the certificate chain that will be verified. Use the default of **5**, or enter the numeric value your configuration requires.
9. **mutual-authenticate**—Define whether or not you want the Net-Net SBC to mutually authenticate the client. Leave this parameters set to its default of **disabled**, or type in **enabled** to turn on mutual authentication.
10. **tls-version**—Enter the TLS version you want to use with this TLS profile. Valid values are **TLSv1**, **SSLv3**, and **compatibility** (default).
11. Save your work.
12. Exit out to the configuration terminal menu to apply the TLS profile.

```
ACMEPACKET(tls-profile)# exit
ACMEPACKET(security)# exit
ACMEPACKET(configure)#
```

Applying a TLS Profile

To apply the TLS profile, you need to specify it for the SIP interface with which it will be used. You must take this step from within the SIP interface configuration.

1. Type **session-router** and press <Enter> to access the **session-router** path.
ACMEPACKET(configure)# **session-router**
2. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-interface**
ACMEPACKET(sip-interface)#
3. Select the existing SIP interface to which you want to apply the TLS profile. If you do not know the name of the profile, press Enter again after you use the select command to see a list of all SIP interfaces. Type in the number corresponding to the SIP interface you want to select, and press <Enter>. You will then be modifying that SIP interface.
ACMEPACKET(sip-interface)# **select**
4. Type **sip-ports** and <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-interface)# **sip-ports**
ACMEPACKET(sip-port)#
5. Type **select** and the number of the pre-configured sip interface you want to configure.
ACMEPACKET(sip-interface)# **select 1**
6. **transport-protocol**—Change the transport protocol to TLS.
ACMEPACKET(sip-interface)# **transport-protocol tls**
7. **tls-profile**—Enter the name of the TLS profile you want applied. This is the same value you enter for the name parameter in the TLS profile configuration. This profile will be applied when the transport protocol is TLS.
ACMEPACKET(sip-interface)# **tls-profile acmepacket**
8. Save your updated SIP interface configuration.

Reusing a TLS Connection

The Net-Net SBC supports TLS connection reuse if and when an **alias** is included in the Via header by the originator of the TLS connection. When this is the case, the Net-Net SBC reuses the same connection for any outgoing request from the Net-Net SBC.

Keeping Pinholes Open at the Endpoint

The Net-Net SBC provides configurable TCP NAT interval on a per-realm basis. You need to configure a NAT interval for the applicable realm to support either all conforming or all non-conforming endpoints.

- Conforming endpoints use the draft-jennings sipping-outbound-01. It describes how to keep the endpoint keeps the connection alive.

Note: Currently the endpoint uses REGISTER.

- Non-conforming endpoints have short NAT interval, where the HNT application with the TCP connection for TLS operates as it does for regular TCP. We give the UA a shorter expires time so that it refreshes frequently, implicitly forcing the UA to keep the TVP socket open and reuse it for further requests (in-dialog or out-of-dialog). Regular requests using TLS sent from the Net-Net SBC to the UA reuse the same TCP connection so that further TLS certificate exchanges are not required.

Viewing Certificates

You can view either a brief version or detailed information about the certificates. The show security certificates command has the following syntax

```
show security certificate [brief | details | list | pem]
```

Brief Version

Obtaining the brief version uses this syntax, and will appear like the following example:

```
ACMEPACKET# show security certificate brief acmepacket
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 24 (0x18)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=US, ST=ma, L=burlington, O=acme, OU=pki, CN=acme test
root ca 002 no liability
    Validity
      Not Before: Mar 24 17:40:08 2008 GMT
      Not After : Mar 24 17:40:08 2009 GMT
    Subject: C=US, ST=MA, L=Burlington, O=acme, OU=pki, CN=jenova-
003
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      Netscape Cert Type:
        SSL Server
      Netscape Comment:
        acmepacket generated certificate
      X509v3 Subject Key Identifier:
        62:81:53:47:EA:BC:11:5D:1E:02:61:10:F2:2E:3D:A7:E8:79:3D:0E
      X509v3 Authority Key Identifier:
```

```
keyid:BC:BA:43:8F:C1:15:54:D4:4B:63:43:6E:EF:22:58:66:A9:00:35:E1
    DirName:/C=US/ST=ma/L=burlington/O=acme/OU=pki/CN=acme
test root ca 002 no liability
    serial:00
```

```
X509v3 Issuer Alternative Name:
    <EMPTY>
```

```
X509v3 Subject Alternative Name:
    <EMPTY>
```

Certificate Request:

```
Data:
    Version: 0 (0x0)
    Subject: C=US, ST=MA, L=Burlington, O=acme, OU=pki, CN=jenova-
003
    Attributes:
        X509v3 Key Usage          :digitalSignature,keyEncipherment
```

Detailed Version

Obtaining the detailed version uses this syntax, and will appear like the following example:

```
ACMEPACKET#show security certificate details acmepacket
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 24 (0x18)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=ma, L=burlington, O=acme, OU=pki, CN=acme test
root ca 002 no liability
        Validity
            Not Before: Mar 24 17:40:08 2008 GMT
            Not After : Mar 24 17:40:08 2009 GMT
        Subject: C=US, ST=MA, L=Burlington, O=acme, OU=pki, CN=jenova-
003
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
            Modulus (2048 bit):
                00:de:b6:e9:69:64:ea:d8:31:6e:b6:ca:c9:ca:77:
                43:d3:93:b4:82:25:20:b2:c0:cc:12:a8:7e:81:bc:
                e3:e4:2f:ac:10:bb:33:fc:73:22:19:3e:7b:57:20:
                6a:22:55:b5:c9:73:78:00:8e:d2:64:48:25:24:1d:
                53:e4:f9:95:e8:0e:8d:d0:11:85:ab:78:61:7a:e0:
                3b:c0:a8:a6:09:79:b6:e8:84:36:0b:9b:2d:51:f9:
                9c:f1:d2:23:0a:a8:3a:2a:fa:ea:e7:b1:65:86:e6:
                c3:d7:89:0e:2f:f4:ea:42:a4:6d:6e:89:cc:c1:51:
                02:4b:59:de:fb:5a:05:c1:7b:ca:bf:ce:5c:86:ae:
                8f:a2:dc:e5:7a:14:75:b5:a2:1e:f0:87:b0:a6:98:
                58:96:da:e1:78:68:34:50:07:38:aa:18:0e:24:f5:
                93:be:78:37:9f:d2:ad:28:80:00:40:7b:1b:9a:97:
                e6:54:2e:8f:46:a0:52:7c:3f:20:fa:27:58:60:66:
                8b:c3:86:10:ba:b4:dc:4c:ca:9c:4b:01:6f:95:e7:
                bf:42:ee:77:86:09:5f:ab:7f:09:91:e6:b8:50:13:
```

```

74:9f:8f:c4:44:a2:5f:98:39:b5:a0:44:fd:76:9f:
ed:c3:ee:0f:e3:a6:f6:6f:7f:96:65:5d:60:96:c1:
72:f5
Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  Netscape Cert Type:
    SSL Server
  Netscape Comment:
    acmepacket generated certificate
  X509v3 Subject Key Identifier:

62:81:53:47:EA:BC:11:5D:1E:02:61:10:F2:2E:3D:A7:E8:79:3D:0E
  X509v3 Authority Key Identifier:

keyid:BC:BA:43:8F:C1:15:54:D4:4B:63:43:6E:EF:22:58:66:A9:00:35:E1
  DirName:/C=US/ST=ma/L=burlington/O=acme/OU=pki/CN=acme
test root ca 002 no liability
  serial:00

  X509v3 Issuer Alternative Name:
    <EMPTY>

  X509v3 Subject Alternative Name:
    <EMPTY>

Signature Algorithm: sha1WithRSAEncryption
8a:65:53:22:e2:97:87:32:77:e4:82:cd:c1:0e:fe:88:5a:e0:
06:24:fe:f9:3f:9f:b8:c4:54:b1:cc:be:da:26:61:b5:36:42:
bc:f2:37:fe:79:22:91:d8:dc:9b:e0:30:59:e4:47:86:32:b9:
2e:41:ec:a0:81:f6:fe:d7:24:69:0f:ef:fb:d2:81:4d:ef:93:
99:6a:01:20:2e:33:fe:e6:62:02:93:6c:41:91:1d:cf:91:85:
03:d6:80:00:e0:36:7a:69:c7:ea:93:c3:6a:5d:a8:1c:7b:2b:
c4:99:ff:5c:1e:2b:9b:e7:7c:bf:c7:03:3f:cc:94:9a:42:c0:
a9:e3:f8:45:a7:0e:46:cc:f8:46:5c:52:b7:c9:34:06:01:b7:
b9:7b:20:79:5f:a1:50:ec:c8:5c:90:65:10:be:71:4d:17:66:
5a:ac:ea:c9:77:9b:1a:86:7f:2d:2f:f0:8b:5e:cc:4b:fe:bf:
cf:8e:31:d1:75:79:4a:f5:47:7a:45:9e:07:c5:ef:f7:5d:3b:
c5:3d:8e:fd:2a:25:41:5c:b2:1a:0b:6f:5d:87:f0:67:b4:90:
65:10:c9:b0:fd:d0:f6:96:a9:d5:1a:07:d4:9b:95:e4:e3:98:
0f:a2:4c:aa:d7:6b:24:e4:3c:d7:d2:a3:e0:66:52:eb:b0:73:
9d:49:ff:4f

Certificate Request:
  Data:
    Version: 0 (0x0)
    Subject: C=US, ST=MA, L=Burlington, O=acme, OU=pki, CN=jenova-
003
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
      Modulus (2048 bit):
        00:de:b6:e9:69:64:ea:d8:31:6e:b6:ca:c9:ca:77:
        43:d3:93:b4:82:25:20:b2:c0:cc:12:a8:7e:81:bc:

```

```

e3:e4:2f:ac:10:bb:33:fc:73:22:19:3e:7b:57:20:
6a:22:55:b5:c9:73:78:00:8e:d2:64:48:25:24:1d:
53:e4:f9:95:e8:0e:8d:d0:11:85:ab:78:61:7a:e0:
3b:c0:a8:a6:09:79:b6:e8:84:36:0b:9b:2d:51:f9:
9c:f1:d2:23:0a:a8:3a:2a:fa:ea:e7:b1:65:86:e6:
c3:d7:89:0e:2f:f4:ea:42:a4:6d:6e:89:cc:c1:51:
02:4b:59:de:fb:5a:05:c1:7b:ca:bf:ce:5c:86:ae:
8f:a2:dc:e5:7a:14:75:b5:a2:1e:f0:87:b0:a6:98:
58:96:da:e1:78:68:34:50:07:38:aa:18:0e:24:f5:
93:be:78:37:9f:d2:ad:28:80:00:40:7b:1b:9a:97:
e6:54:2e:8f:46:a0:52:7c:3f:20:fa:27:58:60:66:
8b:c3:86:10:ba:b4:dc:4c:ca:9c:4b:01:6f:95:e7:
bf:42:ee:77:86:09:5f:ab:7f:09:91:e6:b8:50:13:
74:9f:8f:c4:44:a2:5f:98:39:b5:a0:44:fd:76:9f:
ed:c3:ee:0f:e3:a6:f6:6f:7f:96:65:5d:60:96:c1:
72:f5
Exponent: 65537 (0x10001)
Attributes:
    X509v3 Key Usage          :digitalSignature,keyEncipherment
Signature Algorithm: sha1WithRSAEncryption
d9:6f:b7:5d:97:01:c3:83:32:2a:60:44:9a:7f:c9:5f:92:b8:
01:cf:8c:47:38:a1:cf:24:74:de:e0:3b:5a:09:69:9b:35:a6:
60:60:4f:d0:46:40:19:a2:32:26:3c:c2:c3:88:12:e8:4b:4d:
41:33:18:23:83:f2:fa:3d:09:69:ab:38:df:25:8a:40:b6:e7:
3b:b4:51:cc:0e:2a:5a:5e:1e:6c:26:dc:fb:6d:ea:89:e1:bb:
14:f1:2b:1b:ff:be:5b:27:14:5f:b8:22:e8:58:96:21:5b:26:
55:fb:9f:2f:7c:ce:19:09:bb:3f:8c:bf:99:97:18:9a:9a:cb:
d3:0d:3d:59:a8:8c:c5:fb:bb:c5:ea:27:00:94:b4:44:6a:30:
2d:3a:c0:36:ca:82:88:ce:1f:ba:9f:ff:da:a4:60:13:70:37:
2a:c7:d7:44:e2:94:c0:20:11:13:c3:26:ae:49:39:32:2c:57:
29:79:64:e4:44:69:b6:5c:30:65:1d:d4:74:3f:f6:f8:79:f7:
fd:9a:88:f7:c0:a2:67:52:69:1f:ad:bb:d5:db:3d:29:41:f9:
45:0e:ff:02:41:4b:73:ab:b2:f4:27:66:40:c7:f9:ef:e7:c6:
a1:c4:77:2d:6b:c8:c3:68:77:1b:c2:ae:f9:c0:ca:00:9f:b7:
b5:c0:1d:50

```

Display Certificate in PEM Format

The following command will display the certificate in PEM format:

```
ACMEPACKET#show security certificate pem test
```

```
-----BEGIN CERTIFICATE-----
```

```

MIIEhTCCA22gAwIBAgIBGDANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEL
MAkGA1UECBMcbWExEzARBgNVBAcTCmJ1cmxpbmd0b24xDTALBgNVBAoTBGfjbWUx
DDAKBgNVBAsTA3BraTcrMkGA1UEAxMiYWNTZSB0ZXN0IHJvb3QgY2EgMDAyIG5v
IGxpYWJpbGleTAEwFw0wODAzMjQxNzQwMDhAFw0wOTAzMjQxNzQwMDhAMGExCzAJ
BgNVBAYTAlVTMQswCQYDVQQIEwJNQTETMBEGA1UEBxMKQnVybGluZ3R3b3R3b3R3
A1UECHMEYWNTZTEMAoGA1UECXMdGtpMRMWEQYDVQQDEwppqZW5vdmdETMDAzMIIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3rbpaWTq2DFutsrJyndD0500
giUgssDMEqh+gbzj5C+sELsz/HMiGT57VyBqI1w1yXN4AI7SZEglJB1T5PmV6A6N
0BGFq3hheuA7wKimCXm26IQ2C5stUfmc8dIjCqg6Kvrq57F7hubD14k0L/TqQqRt
bonMwVECS1ne+1oFwXvKv85chq6PotzlehR1taIe8IewpphYlrrheGg0UAc4qhG0

```

```
JPWTVng3n9KtKIAAQHsbmpfmVC6PRqBSfD8g+idYYGaLw4YQurTcTMqcSwFvlee/
Qu53hg1fQ38Jkea4UBN0n4/ERKJfmDm1oET9dp/tw+4P46b2b3+WZV1glsFy9QID
AQABo4IBLjCCASowCQYDVR0TBAlwADARBg1ghkgBhvCAQEEBAMCBkAwLwYJYIZI
AYb4QgENBCIWIGFjbWVwYWNrZXQgZ2VuZXJhdGVkIGNlcnRpZmljYXRlMB0GA1Ud
DgQWBBRigVNH6rWRXR4CYRDyLj2n6Hk9DjCBowYDVR0jBIGbMIGYgBS8ukOPwRVU
1EtjQ27vIlhmQQA14aF9pHsweTELMaKGA1UEBhMCVVMxCzAJBgNVBAGTAM1hMRMw
EQYDVQQHEWpidXJsaw5ndG9uMQ0wCwYDVQQKEWRhY21lMQwwCgYDVQQLEWNa2kx
KzApBgNVBAMTImFjbWUgdGVzdCBYb290IGNhIDAwMiBubyBsaWwpcHMAQAw
CQYDVRSBAIwADAJBgNVHREEAjAAMA0GCsGSIb3DQEBAQUAA4IBAQCkZVMi4peH
Mnfkgs3BDv6IWuAGJP75P5+4xFSxZL7aJmG1NkK88jf+eSKR2Nyb4DBZ5EeGMrku
Qeyggfb+1yRpD+/70oFN750ZagEgLjP+5mICk2xBkR3PkYUD1oAA4DZ6acFqk8Nq
XagceyvEmf9chiub53y/xwM/zJSaQsCp4/hFpw5GzPhGXFk3yTQGAbe5eyB5X6FQ
7MhckGUQvnFNF2ZarOrJd5sahn8tL/CLXsXl/r/PjjHRdXlK9Ud6RZ4Hxe/3XTvF
PY79KiVBXLIaC29dh/BntJB1EMmw/dD2lqnVGgfUm5Xk45gPokyq12sk5DzX0qPg
ZLLrsh0dsf9P
```

-----END CERTIFICATE-----

Certificates List

The show security certificate list command lists all installed certificates. For example:

```
ACMEPACKET# show security certificate list
1: acmepacket
```

TLS Cache Statistics

You can view the number of entries in the TLS session cache, and then clear this cache as needed.

To view number of entries in the TLS session cache use the following command:

```
show security tls cache-count
```

To clear the entries in the TLS session cache use the following command:

```
security tls cache-clear
```

Denial of Service for TLS

This section explains the DoS for TLS feature. With this feature, the Net-Net SBC can provide protection from TCP/TLS message flood by limiting the number of connections from an end point and by limiting the number of simultaneous TCP/TLS connections to a SIP interface.

The Net-Net SBC protects against a flood of invalid TLS messages and against end points establishing TCP/TLS connections or doing an initial registration without then sending any messages. The Net-Net SBC protects against:

- Too many simultaneous TLS connections being requested by a single IP address by limiting the number of TLS connections from a single IP address. There is a maximum simultaneous number of TCP/TLS connections a SIP interface will allow from a single IP address.
- Too many simultaneous TLS connections being requested by limiting the maximum number of connections for a SIP interface. There is a maximum number of simultaneous TCP/TLS connections a SIP interface will allow in aggregate from all IP addresses served by that signaling interface.

- End points establishing TCP/TLS connections without then sending any messages (application layer messages post TLS handshake complete). Triggered by inactivity as measured by lack of any message from this peer.
- End points doing an initial registration without then sending any messages.
This timer could be used by the administrator to detect errors with the SIP configuration. It is expected that whenever an end point establishes a TCP/TLS connection, the end point will keep the connection active by sending messages with REGISTER or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level ERROR is generated.)
- Malformed packets by counting and limiting the maximum number of malformed packets. Whenever an invalid TLS message is received, the internal counter corresponding to invalid-signal-threshold is incremented. When the invalid signal threshold reaches the configured value, the end point will be denied for the configured deny period. (Also requires configuration of the tolerance window in media manager.)

ACLI Instructions and Examples

You configure the SIP interface and the realm to support DoS for TLS.

Configuration the SIP Interface

To configure the SIP interface:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the system-level configuration elements.


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


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.
4. **max-incoming-conns**—Enter the maximum number of simultaneous TCP/TLS connections for this SIP interface. The range of values is 0 to 40000. The default value of 0 disables this parameter.
5. **per-src-ip-max-incoming-conns**—Enter the maximum number of connections allowed from an end point. The range of values is 0 to 40000. The default value of 0 disables this parameter.

Note: To make this parameter effective, you need to set the realm's access-control-trust-level to low or medium.
6. **inactive-conn-timeout**—Enter the time in seconds you want a connection from an endpoint discontinued. This provides protection from end points doing an initial registration without sending any messages. The default value is zero (0), which disables the timer.
7. Save and activate your configuration.

Configuring the SIP Configuration

To configure the SIP configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-config** and press <Enter>. The system prompt changes.

```
ACMEPACKET(session-router)# sip-config
```

```
ACMEPACKET(sip-config)#
```

From this point, you can configure SIP configuration parameters. To view all sip-config parameters, enter a ? at the system prompt.

4. **inactive-dynamic-conn**—Enter the time in seconds after which if the peer does not send SIP messages after it initiates a TCP connection, the connection is torn down. This protects against endpoints establishing TCP/TLS connections and then not sending messages. The range of values is 0 to 999999999. The default value is 0, which disables the timer.

Because the Net-Net SBC first establishes a TCP connection, then the TLS connection it waits twice the value entered here after the initiation of a TLS connection before tearing down the connection.

After an endpoint establishes a TCP/TLS connection, it is supposed to keep the connection active by sending messages or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level "ERROR" is generated.

Configuring the Realm

To configure the realm:

1. In Superuser mode, type **configure terminal** and press <Enter>.
2. Type **media-manager** and press <Enter> to access the media-related configurations.

```
ACMEPACKET# configure terminal
```

```
ACMEPACKET(configure)# media-manager
```

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

```
ACMEPACKET(media-manager)# realm-config
```

```
ACMEPACKET(realm-config)#
```

From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.

4. **deny-period**—Indicate the time period in seconds after which the entry for this host is removed from the deny list. Value range is:
 - minimum: zero (0) seconds
 - maximum: 4294967295 seconds

The default value is 30 seconds.

5. **invalid-signal-threshold**—Enter the maximum number of simultaneous TCP/TLS connections for this SIP interface.

Setting this parameter provides protection from flood of invalid TLS messages. Whenever an invalid TLS message is received, the internal counter is

incremented. When the invalid signal threshold reaches the configured value, the end point will be denied for the configured deny period.

6. **access-control-trust-level**—Set the trust level for the host within the realm. The available values are:
 - none: host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - low: host can be promoted to the trusted list or demoted to the deny list.
 - medium: host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - high: host is always trusted.
7. Save and activate your configuration.

TLS Session Caching

TLS session caching provides the ability for the Net-Net SBC to cache key information for TLS connections, and to set timer for the length of time that the information will be cached.

Without this feature enabled, when the Net-Net SBC and a TLS client perform the handshake portion of the authentication sequence, they exchange a shared secret and encryption keys are generated. And one of the results of the successful handshake is the creation of a unique session identifier. In the event that an established TLS connection is torn down but the client wants to reinstate it, this entire process must be repeated. Since the process is resource-intensive, the TLS session caching feature can be enabled to avoid repeating the handshake process for previously authenticated clients—thereby preserving valuable Net-Net SBC resources.

If TLS session caching is enabled on the Net-Net SBC, a formerly authenticated client can request re-connection using the unique session identifier from the previous session. The Net-Net SBC checks its cache, finds the session identifier, and reinstates the client. This reduces the handshake process to three messages. However, should the client offer an invalid session identifier (one that the Net-Net SBC has never seen, or one that has been deleted from its cache) the re-connection will fail, and will need to be negotiated from scratch.

ACLI Instructions and Examples

TLS session caching is global for all TLS functions on your Net-Net SBC. A new global TLS configuration (**tls-global**) has been added to the system for this purpose.

To enable global TLS session caching:

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


```
ACMEPACKET# configure terminal
```
2. Type **security** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```
3. Type **tls-config** and press <Enter>.


```
ACMEPACKET(security)# tls-config
ACMEPACKET(tls-config)#
```

4. **session-caching**—Set the state for TLS session caching to **enabled** if you want to turn this feature on. The default is **disabled**.
5. **session-cache-timeout**—Enter the time in hours that you want the Net-Net SBC to cache unique session identifiers so that previously authenticated clients can reconnect. The default value is **12**; the minimum value is **0**, and the maximum is **24**.

If you set this parameter to **0**, then cache entries will never age. RFC 2246, “The TLS Protocol Version 1.0,” recommends that you set this parameter at the maximum, **24**.

Untrusted Connection Timeout for TCP and TLS

You can configure the Net-Net SBC for protection against “starvation attacks” for socket-based transport (TCP or TLS) for SIP access applications. During such an occurrence, the attacker would open a large number of TCP/TLS connections on the Net-Net SBC and then keep those connections open using SIP messages sent periodically. These SIP messages act as keepalives, and they keep sockets open and consume valuable resources.

Using its ability to promote endpoints to a “trusted” status, the Net-Net SBC now closes TCP/TLS connections for endpoints that do not enter the trusted state within the period of time set for the untrusted connection timeout. The attacking client is thus no longer able to keep connections alive by sending invalid messages.

This feature works by setting a value for the connection timeout, which the Net-Net SBC checks whenever a new SIP service socket for TCP or TLS is requested. If the timer’s value is greater than zero, then the Net-Net SBC starts it. If the timer expires, then the Net-Net SBC closes the connection. However, if the endpoint is promoted to the trusted state, then the Net-Net SBC will cancel the timer.

Caveats

This connection timeout is intended for access applications only, where one socket is opened per-endpoint. This means that the timeout is not intended for using in peering applications; if this feature were enabled for peering, a single malicious SIP endpoint might cause the connection to be torn down unpredictably for all calls.

ACLI Instructions and Examples

The untrusted connection timer for TCP and TLS is set per SIP interface.

To set the untrusted connection timer for TCP and TLS:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>.


```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

If you are adding support for this feature to a pre-existing SIP configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.

4. **untrusted-conn-timeout**—Enter the time in seconds that you want the Net-Net SBC to keep TCP and TLS connections open for untrusted endpoints. The default for this parameter is 0, which will not start the timer.
5. Save and activate your configuration.

TLS Statistics and Configuration

You can view listening and connected TLS sockets with the **show sip sockets** or **show sfe sockets** commands. Displaying specific TLS socket statistics with the **show sfe sockets <socket number>** command looks like this:

```
ACMEPACKET# show sfe sockets 23
Server State                Active
-----
09:03:16-100 [23] sipt@0.1.0 TLS[0:0/0]172.16.0.241:5061
<=172.16.0.10:55389

----- Lifetime -----
Recent      Total    PerMax
Sent         4         4         4
Bytes Sent   2033      2033      2033
Recieved     1         1         1
Bytes Recieved 313       313       313
Loop Back - Same Sock 0          0          0
Loop Back - Other Sock Recv 0          0          0
Loop Back - Other Sock Sent 0          0          0
Errors       0          0          0
-----

Responder, New, TLSv1/SSLv3, DHE-RSA-AES256-SHA
TLS Profile: t4
SSL: read 718 bytes, written 2033 bytes
Client Certificate: not available
SSL-Session:
  Protocol   : TLSv1
  Cipher     : DHE-RSA-AES256-SHA
  Session-ID:
  Session-ID-ctx:
  Master-Key:
8ecafbe6c579b27aca2f3105a7ae06dbe88a65adb5cccc7a72da9324760f6ca3
407f298b7c805ed793080c3957a
  Key-Arg    : None
  Start Time: 1190883768
  Timeout    : 300 (sec)
  Verify return code: 0 (ok)
-----
```

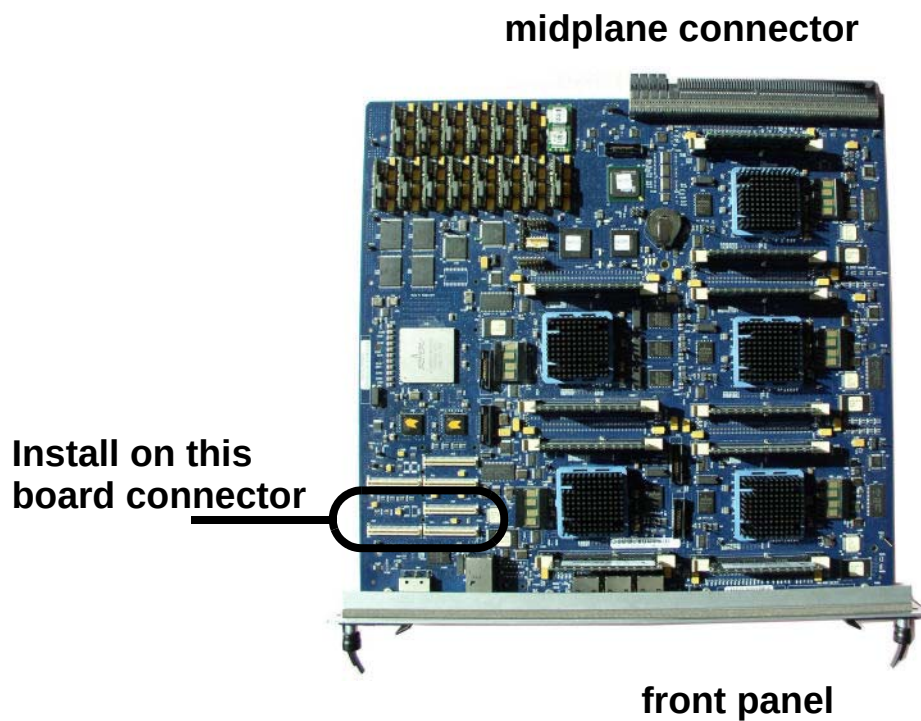
TLS Hardware Support

TLS uses a Signaling Security Module (SSM) for hardware-accelerated encryption, decryption, and random media generation. This SSM must be installed according to the image contained in this section. Failure to install the SSM correctly will render the TLS feature unusable.

Follow all service procedures through the “Processing and Interface Unit Maintenance and Installation” section in the System Maintenance chapter in the *Net-Net 9200 Hardware Installation Guide*.

The following photograph shows an SPU. The SPU's orientation is noted in the image. The SSM is installed in the lower left corner of the SPU.

You must install the SSM in the board connector circled below.



IPsec Support

The Net-Net SBC offers hardware-based IPsec for securing signaling and media at the network layer. This feature is supported by a 4-port SFP GigE IPsec-accelerated NIU on the Net-Net 9200.

In Net-Net OS Release D6.0 and above, IPsec functionality is limited to 1000 tunnels.

Supported Protocols

The Net-Net SBC's IPsec implementation supports all required tools for securing Internet communication via the IPsec protocol suite. The following paragraphs list and explain the protocols within the IPsec suite that the Net-Net SBC supports. This chapter does not explain how to design and choose the best protocols for your application.

AH vs. ESP

The Net-Net SBC supports the two encapsulations that IPsec uses to secure packet flows. Authentication Header (AH) is used to authenticate and validate IP packets. Authentication means that the packet was sent by the source who is assumed to have sent it. Note that AH is incompatible with NAT. Validation means that the recipient is assured that the packet has arrived containing the original, unaltered data as sent.

ESP (Encapsulating Security Payload) provides AH's authentication and validations and extends the feature set by ensuring that the IP packet's contents remain confidential as they travel across the network. Using an encryption algorithm that both peers agree upon, ESP encrypts a full IP packet so that if intercepted, an unauthorized party cannot read the IPsec packet's contents.

Tunnel Mode vs. Transport Mode

In addition to its security encapsulations, the IPsec suite supports two modes: tunnel mode and transport mode. Tunnel mode is used most often for connections between gateways, or between a host and a gateway. Tunnel mode creates a VPN-like path between the two gateways and encapsulates the entire original packet. Transport mode is used to protect end-to-end communications between two hosts providing a secured IP connection and encrypts just the original payload.

Cryptographic Algorithms

IPsec works by using a symmetric key for validation and encryption. Symmetric key algorithms use the same shared secret key for encoding and decoding data on both sides of the IPsec flow. The Net-Net SBC's IPsec feature supports the following encryption algorithms:

- 3DES
- AES128CBC
- AES256CBC
- AES128CTR
- AES256CTR

The Net-Net SBC can quickly generate keys for all of the above mentioned algorithms from the CLI. It can additionally support HMAC-SHA1 or HMAC-MD5 keyed-hash message authentication codes. Only manual keying is currently supported for both hash authentication and data encryption. Therefore, all keys must be provisioned on the Net-Net SBC by hand.

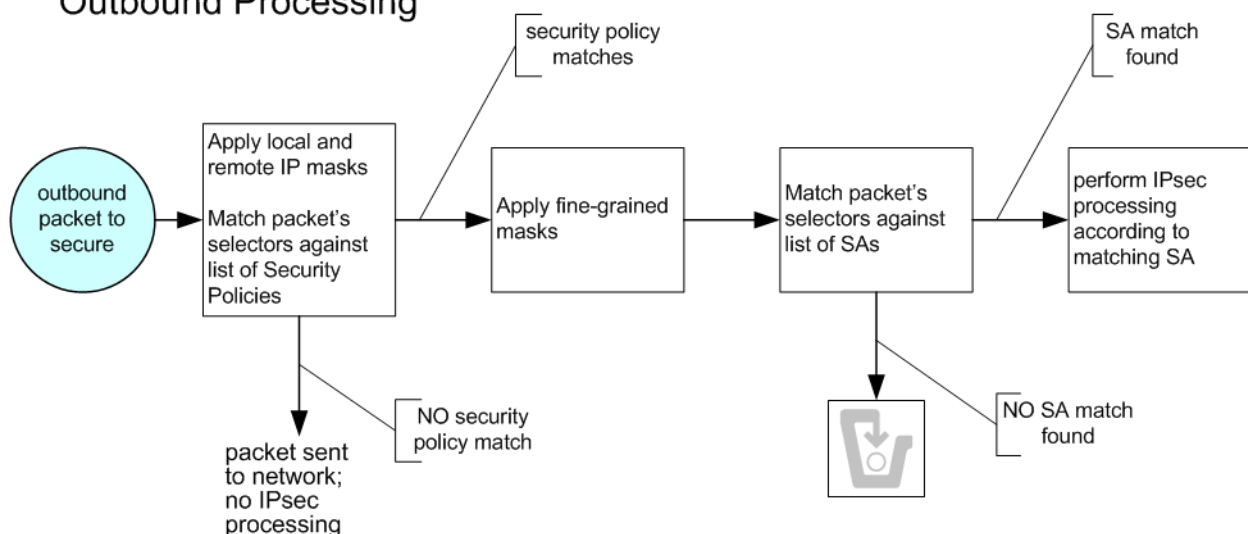
IPsec Implementation

The Net-Net SBC uses separate logic for processing IPsec packets based on whether the traffic is inbound or outbound. The configuration is divided into two pieces, the security policy and the security association (SA). Both the SA and security policies have a directional attribute which indicates if they can be used and/or reused for inbound and outbound traffic.

Outbound Packet Processing

The following diagrams show the steps the Net-Net SBC follows when processing outbound IPsec traffic. Details of each step are described in the following sections.

Outbound Processing



Security Policy

The Net-Net SBC first performs a policy lookup on outbound traffic to test if it should be subjected to IPsec rules. A security policy, local policy applicable for IPsec functionality, defines the matching criteria for outgoing network traffic to secure. It is configured on a network interface basis.

Configuring a security policy is similar to a local policy, with additional IPsec-specific parameters. Unlike a local policy, used for routing, a security policy is used for packet treatment. As with a local policy, a set of selector values is matched against the outbound flow's following characteristics:

- VLAN
- Source IP address (plus mask)
- Source IP port
- Destination IP address (plus mask)
- Destination IP port
- Transport Protocol

Each of these selection criteria can be defined by a wildcard except for the VLAN ID, which can be ignored. This flexibility aids in creating selection criteria that ranges from highly restrictive to completely permissive. In addition to the main traffic matching criteria, a priority parameter is used to prioritize the order that configured security policies are checked against. The #0 policy is checked first, #1 policy is checked next, continuing to the lowest prioritized policy being checked last.

Once the outbound traffic matches a policy, the Net-Net SBC proceeds to the next step of outbound IPsec processing. If no matching security policy is found, the default pass-through policy allows the packet to be forwarded to the network without any security processing. A maximum of 126 configurable security policies are available.

Fine-grained policy Selection

After a positive match between outbound traffic and the configured selectors in the security policy, the Net-Net SBC can perform a calculation between a set of fine-grained packet selectors and the outbound packet. The fine-grained policy masking criteria are:

- Source IP subnet mask
- Source port mask
- Destination IP subnet mask
- Destination port mask
- VLAN mask
- Transport protocol mask

By default, the fine-grained security policy is set to match and pass all traffic untouched to the security association (SA) portion of IPsec processing.

Fine-grained policy selection works by performing a logical AND between outbound traffic's fine-grained selectors and the traffic's corresponding attributes. The result is then used to find the matching SA. Applying a fine-grained mask has the effect of forcing a contiguous block of IP addresses and/or ports to appear as one address and or port. During the next step of IPsec processing, when an SA is chosen, the Net-Net SBC in effect uses one SA lookup for a series of addresses. Without fine-grained policy selection, unique SAs must always be configured for outbound packets with unique security policy selectors.

Security Associations

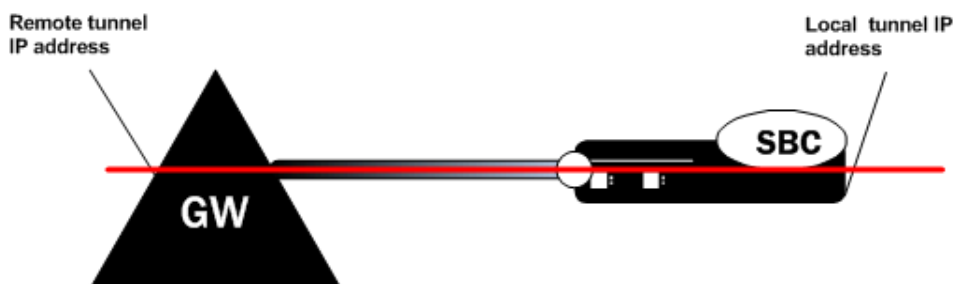
After the Net-Net SBC determines that outgoing traffic is subject to IPsec processing, and applies fine-grained masking, an SA lookup is performed on the traffic. An SA is the set of rules that define the association between two endpoints or entities that create the secured communication. To choose an SA, the Net-Net SBC searches for a match against the outgoing traffic's SA selectors. SA selectors are as follows:

- VLAN
- Source IP address
- Source IP port
- Destination IP address
- Destination IP port
- Transport Protocol

If there is a match, the Net-Net SBC secures the flow according to security parameters defined in the SA that the Net-Net SBC chooses. The packet is then forwarded out of the Net-Net SBC. If no match is found, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG.

Secure Connection Details

Several parameters define an IPsec connection between the Net-Net SBC and a peer. When planning an IPsec deployment, the primary architectural decisions are which IPsec protocol and mode to use. The two choices for IPsec protocol are ESP or AH, and the two choices for IPsec mode are either tunnel or transport. IPsec protocol and mode are both required for an SA configuration. When creating an IPsec tunnel (tunnel mode), the SA must also define the two outside IP addresses of the tunnel.

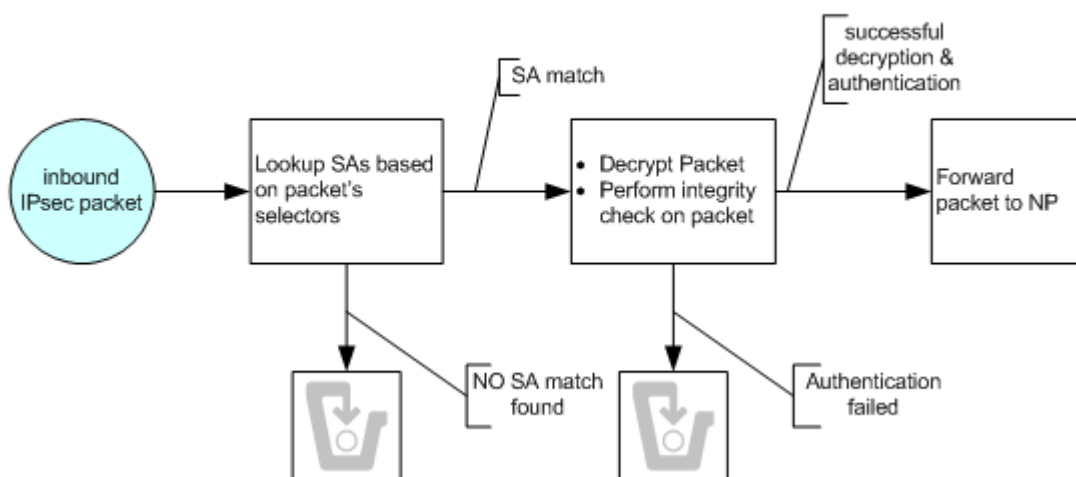


The authentication algorithm and the authentication key must always be configured. The Net-Net SBC supports hmac-md5 or hmac-sha1 authentication algorithms. Because only manual keying is supported, the key must be entered by hand. When encryption is required, the encryption algorithm and the encryption key must be configured. The Net-Net SBC supports 3des, aes-128-cbc, aes-256-cbc, aes-128-ctr, and aes-256-ctr encryption algorithms. When using the two encryption protocols that operate in AES counter mode (RFC 3686), an additional nonce value is required. In addition, the security parameter index (SPI) must be configured for each SA. All SPI values must be unique as well, across all SAs.

Inbound Packet Processing

The following diagram shows the steps the Net-Net SBC follows when processing inbound IPsec traffic. Details of each step are described in the following sections.

Inbound Processing



IP Header Inspection

Processing inbound IPsec packets begins by the Net-Net SBC inspecting an inbound IP packet's headers. If the packet is identified as IPsec traffic, as determined by the presence of an AH or ESP header, an SA policy lookup is performed. If the traffic is identified as non-IPsec traffic, as determined by the lack of an IPsec-type (AH or

ESP) header, it still is subject to a policy lookup. However, due to the default allow policy, the packet is forwarded directly to the NP, without any security processing.

SA Matching

The Net-Net SBC proceeds to match the inbound IPsec traffic's selectors against configured SAs. Inbound selector masking is performed where noted. These selectors are:

- VLAN (plus mask)
- Destination IP address (plus mask)
- SPI

If no matching SA is found, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG. When the Net-Net SBC finds a matching SA, the packet is authenticated and decrypted according to the configuration and sent to the Net-Net SBC's NP for continued processing.

HA Considerations

Anti-replay mechanisms, running on IPsec peers, can cause instability with the Net-Net SBCs configured in an HA pair. The anti-replay mechanism ensures that traffic with inconsistent (non-incrementing) sequence numbers is labeled as insecure, assuming it could be part of a replay attack. Under normal circumstances, this signature causes the remote endpoint to drop IPsec traffic.

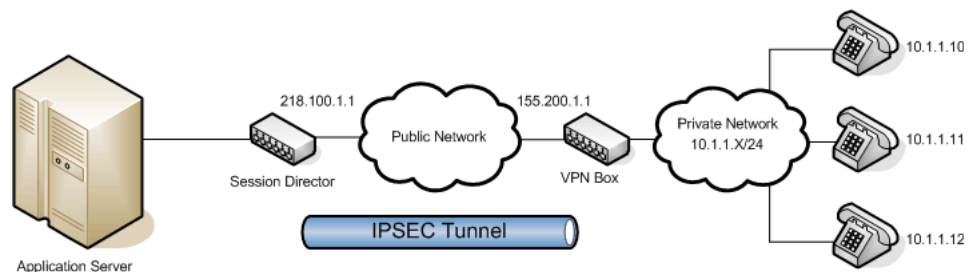
When a failover occurs between HA peers, the newly-active system starts sending traffic with the IPsec sequence number starting at 0. A remote system's anti-replay mechanism observes this and labels the traffic as insecure. It is therefore recommended that anti-replay protection not be used with Net-Net SBCs in an HA configuration. This situation does not create any problems as long as IPsec peers are not configured to use anti-replay mechanisms.

Packet Size Considerations

The security processor supports receipt of jumbo frames up to 9K (9022 bytes with VLANs, 9018 without). Under normal operation the default outgoing maximum packet size of 1500 bytes is used. This packet size includes the IPsec headers, which will result in less space for packet data (SIP signaling, RTP, etc...).

IPsec Application Example

In this example, the Net-Net SBC terminates an IPsec tunnel. The remote side of the tunnels is a dedicated VPN appliance in the public Internet. Behind that VPN appliance are three non-IPsec VoIP phones. In this scenario, the VPN box maintains the IPsec tunnel through which the phones communicate with the Net-Net SBC.



Without the fine-grained option (or alternatively IKE), an SA entry would need to be configured for each of the three phones, communicating over the IPsec tunnel (resulting in 3 tunnels).

This does not scale for manual-keying with a large number of endpoints. Using the fine-grained configuration allows any number of endpoints on the 10.1.1.X network to use a single security association (a coarse-grain configuration). The configuration in this example follows:

A packet sent from the Net-Net SBC to any of the phones will match the policy pol1. The remote-ip-mask parameter of the fine-grained configuration will then be masked against the remote-ip, resulting in a SA selector value of 10.1.1.0. This matches security-association sa1, and the packet will be secured and sent over the tunnel. The tunnel-mode addresses in the security-association represent the external, public addresses of the termination points for the IPsec tunnel.

Packets returning from the 10.1.1.0 side of the IPsec tunnel will first match the tunnel-mode local-ip-addr of 218.100.1.1. The packets will then be decrypted using the SA parameters, and the tunneled packet will be checked against the remote-ip-addr field of the SA.

If the fine-grained mask had not been modified, three discrete SAs would have to be configured: one for each of the three phones.

```
ACMEPACKET(manual)# show
manual
      name                sa1
      spi                 257
      network-interface    M10:0
      local-ip-addr        218.100.1.1
      remote-ip-addr       10.1.1.0
      local-port           0
      remote-port          0
      trans-protocol       ALL
      ipsec-protocol       esp
      direction            both
      ipsec-mode            tunnel
      auth-algo             hmac-sha1
      encr-algo             aes-128-cbc
      auth-key              <key value encrypted, not shown>
      encr-key
      aes-ctr-nonce         0
      full-policy-match     disabled
      tunnel-mode
          local-ip-addr      218.100.1.1
          remote-ip-addr     155.200.1.1
      last-modified-date
ACMEPACKET(manual)#
```

ACLI Instructions and Examples

The following example explains how to configure IPsec on your Net-Net SBC.

Note: If you change the phy-interface slot and port associated with any SAs or SPDs, the Net-Net SBC must be rebooted for the changes to take effect.

Configuring an IPsec Security Policy

To configure an IPsec security policy:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **security** and press <Enter> to access the security path of the configuration menu.


```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```
3. Type **ipsec** and press <Enter>.


```
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)#
```
4. Type **security-policy** and press <Enter>. The prompt changes to let you know that you can begin configuration.


```
ACMEPACKET(ipsec)# security-policy
ACMEPACKET(security-policy)#
```
5. **name**—Enter a name for this security policy. This parameter is required and has no default.
6. **network-interface**—Enter the network interface and VLAN where this security policy applies in the form: interface-name:VLAN
7. **priority**—Enter the priority number of this security policy. Valid values are 0 - 126.
8. **action**—Enter the action the Net-Net SBC should take when this policy matches outbound IPsec traffic. The choices are:
 - **ipsec**—continue processing as IPsec traffic
 - **allow**—forward the traffic without any security processing
 - **discard**—discard the traffic
 - **direction**—Enter the direction of traffic this security policy can apply to. The choices are:
 - **in**—this security policy is valid for inbound traffic
 - **out**—this security policy is valid for outbound traffic
 - **both**—this security policy is valid for inbound and outbound traffic

To define the criteria for matching traffic selectors for this security policy:
9. **local-ip-addr-match**—Enter the source IP address to match.
10. **remote-ip-addr-match**—Enter the destination IP address to match.
11. **local-port-match**—Enter the source port to match. A value of 0 disables this selector.
12. **remote-port-match**—Enter the destination port to match. A value of 0 disables this selector.
13. **trans-protocol-match**—Enter the transport protocol to match. Valid values are: UDP, TCP, ICMP, or ALL [default]
14. **local-ip-mask**—Enter the source IP address mask, in dotted-decimal notation.
15. **remote-ip-mask**—Enter the remote IP address mask, in dotted-decimal notation.

16. Save your work using the ACLI **done** command.

Defining Outbound Fine-Grained SA Matching Criteria

To define outbound fine-grained SA matching criteria:

1. From within the security policy configuration, type **outbound-sa-fine-grained-mask** and press <Enter>. The prompt changes to let you know that you can begin configuration.
2. **local-ip-mask**—Enter the fine-grained source IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
3. **remote-ip-mask**—Enter the fine-grained destination IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
4. **local-port-mask**—Enter the fine-grained source port mask to apply to outbound IP packets for SA matching. This value is entered as an integer and only masks for one value at a time. A value of 65535 allows for all ports to match.
5. **remote-port-mask**—Enter the fine-grained destination port mask to apply to outbound IP packets for SA matching. This value is entered as an integer and only masks for one value at a time. A value of 65535 allows for all ports to match.
6. **vlan-id-mask**—Enter the fine-grained VLAN mask to apply to outbound IP packets for SA matching. Valid values range from 0x0000 through 0xFFFF.
7. **trans-protocol-mask**—Enter the transport protocol ID mask to apply to outbound IP packets for SA matching. This value is entered as an integer and only masks for one value at a time. A value of 255 allows for all ports to match.
8. Save your work using the ACLI **done** command.

Configuring an IPsec SA

To configure an IPsec SA:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **security** and press <Enter> to access the security path of the configuration menu.


```
ACMEPACKET(configure)# security
ACMEPACKET(security)#
```
3. Type **ipsec** and press <Enter>.


```
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)#
```
4. Type **manual-security-association** and press <Enter>. The prompt changes to let you know that you can begin configuration.


```
ACMEPACKET(ipsec)# manual-security-association
ACMEPACKET(manual-security-association)#
```
5. **name**—Enter a name for this security policy.
6. **network-interface**—Enter the network interface and VLAN where this security policy applies in the form: interface_name:VLAN
7. **direction**—Enter the direction of traffic this security policy can apply to. Valid values are: in, out, both.
8. Continue to the next section to complete this configuration.

Defining Criteria for Matching Traffic Selectors per SA

To define the criteria for matching traffic selectors for this SA:

1. From within the **manual-security-association** portion of the ipsec configuration, you need to set the parameters described in this process.
`ACMEPACKET(manual-security-association)#`
2. **local-ip-addr**—Enter the source IP address to match.
3. **remote-ip-addr**—Enter the destination IP address to match.
4. **local-port**—Enter the source port to match. A value of 0 disables this selector.
5. **remote-port**—Enter the destination port to match. A value of 0 disables this selector.
6. **trans-protocol**—Enter the transport protocol to match. Valid values are: **UDP**, **TCP**, **ICMP**, or **ALL** (default).
7. **spi**—Enter the security parameter index. Valid values range from 256-2302
8. **ipsec-mode**—Enter the IPsec mode of this SA. Your choices are **tunnel** or **transport**.
9. **auth-algo**—Enter the IPsec authentication algorithm for this SA. The choices are:
 - **hmac-md5**
 - **hmac-sha1**
 - **null** (default)
10. **auth-key**—Enter the authentication key for the previously chosen authentication algorithm for this SA.

Note: The specified **auth-key** value will be encrypted in the configuration and will no longer be visible in clear-text.

11. **encr-algo**—Enter the IPsec encryption algorithm for this SA. The choices are:
 - **3des**
 - **aes-128-cbc**
 - **aes-256-cbc**
 - **aes-128-ctr**
 - **aes-256-ctr**
 - **null** (default)
 12. **encr-key**—Enter the encryption key for the previously chosen encryption algorithm for this SA.
- Note:** The specified **encr-key** value will be encrypted in the configuration and will no longer be visible in clear-text.
13. **aes-ctr-nonce**—Enter the AES nonce if **aes-128-ctr** or **aes-256-ctr** were chosen as your encryption algorithm.

Defining Endpoints for IPsec Tunnel Mode

To define endpoints for IPsec tunnel mode:

1. From within the **manual-security-association** portion of the security association configuration, you need to set the parameters described in this process.
`ACMEPACKET(manual-security-association)#`
2. Type **tunnel-mode** and press <Enter>.

```
ACMEPACKET(manual-security-association)# tunnel-mode
ACMEPACKET(tunnel-mode)#
```

3. **local-ip-addr**—Enter the local public IP address which terminates the IPsec tunnel.
4. **remote-ip-addr**—Enter the remote public IP address which terminates the IPsec tunnel.
5. Save your work using the ACLI **done** command.

Key Generation

The **generate-key** command generates keys for the supported encryption or authentication algorithms supported by the Net-Net SBC's IPsec implementation. The generate-key commands generate random values which are not stored on the Net-Net SBC, but are only displayed on the screen. This command is a convenient function for users who would like to randomly generate encryption and authentication keys. The generate-key usage is as follows:

```
security generate-key [hmac-md5 | hmac-sha1 | aes-128 | aes-256 | 3des]
```

IPsec Statistics

The commands listed in this section are used to view IPsec statistics from the security association database and for IPsec hardware. All IPSEC-related show commands are available under the **show security** branch of the ACLI.

Security Association Database Commands

show security ipsec sad

This command shows the security association database entries which are programmed into the IPsec processor. In the case of manual keying, the entries should match that of the running configuration. This command is entered as:

```
show security ipsec sad [network-interface] <brief | verbose>
[selectors]
```

The [selectors] field allows you to specify which SA entries you would like to display according to the following table:

Selector	Description
<enter>	Display all entries
Direction	Direction (IN OUT BOTH), Default: BOTH
dst-addr-prefix	Destination address prefix, Default: match any
dst-port	Destination port, Default: match any
ipsec-protocol	IPsec protocol (AH ESP BOTH), Default: BOTH
Spi	security-policy-index, Default: match any
src-addr-prefix	Source address prefix, Default: match any
src-port	Source port, Default: match any
trans-proto	Transport protocol (UDP TCP ICMP ALL), Default: ALL

For example:

```
ACMEPACKET# show security ipsec sad M10 brief
IPSEC security-association-database for interface 'M10':
```

```
Total number of inbound SA records : 1
Total number of outbound SA records : 1
```

Displaying SA's that match the following criteria -

```
spi           : "any"
direction     : both
ipsec-proto    : "any"
src-addr-prefix : "any"
src-port      : "any"
dst-addr-prefix : "any"
dst-port      : "any"
trans-proto    : ALL
vlan-id       : "any"
```

Inbound, SPI: 257

```
destination-address : 192.168.1.2
vlan-id            : 0
sal-mask           : 3
ipsec-proto        : ESP
sad-index          : 0
encr-algo          : aes-128-cbc
auth-algo          : hmac-sha1
ipsec-mode         : transport
match fields:
  src-ip           : 192.168.1.1
  dst-ip           : 192.168.1.2
  src-port         : 0
  dst-port         : 0
  vlan-id          : 0
  trans-proto      : ALL
mask fields:
  src-ip           : 255.255.255.255
  dst-ip           : 255.255.255.255
  src-port         : 0
  dst-port         : 0
  vlan-id          : 0
  protocol         : 0
```

Outbound, SPI: 257

```
source-address     : 192.168.1.2
destination-address : 192.168.1.1
source-port        : 0
destination-port    : 0
trans-proto        : ALL
vlan-id            : 0
sad-index          : 0
encr-algo          : aes-128-cbc
auth-algo          : hmac-sha1
ipsec-mode         : transport
```


**show security ipsec
statistics sad**

This command shows the security policy database entries which are programmed into the IPsec processor. This command is entered as:

```
show security ipsec spd [network-interface]
```

For example:

```
ACMEPACKET#show security ipsec spd M10
```

Inbound SPD records:

SPD Record, Priority: 0

```

    action                                     : "ipsec"
    match fields:
        local-ip-addr                         : 192.168.1.1
        remote-ip-addr                       : 192.168.1.2
        local-port                           : 0
        remote-port                         : 0
        protocol                             : "all"
        vlan-id                              : 0
    mask fields:
        local-ip-addr                         : 255.255.255.255
        remote-ip-addr                       : 255.255.255.255
        local-port                           : "disabled"
        remote-port                         : "disabled"
        protocol                             : "disabled"
        vlan-id                              : 0

```

Outbound SPD records:

SPD Record, Priority: 0

```

    action                                     : "ipsec"
    match fields:
        local-ip-addr                         : 192.168.1.2
        remote-ip-addr                       : 192.168.1.1
        local-port                           : 0
        remote-port                         : 0
        protocol                             : "all"
        vlan-id                              : 0
    mask fields:
        local-ip-addr                         : 255.255.255.255
        remote-ip-addr                       : 255.255.255.255
        local-port                           : "disabled"
        remote-port                         : "disabled"
        protocol                             : "disabled"
        vlan-id                              : 0
    fine-grained:
        valid                                 : "enabled"
        local-ip-mask                        : 255.255.255.255
        remote-ip-mask                      : 255.255.255.255
        local-port-mask                     : 0
        remote-port-mask                   : 0
        vlan-id-mask                       : 4096
        protocol                           : "ALL"

```

IPsec Hardware Statistics

You can query the IPsec processor for hardware identification or Ethernet counts.

show security ipsec statistics gmac

The **show security ipsec statistics gmac** command displays values for the GMAC interface on an IPsec phy card. You can display either all, or only error, receive, or transmit statistics. This command is entered as:

```
show security ipsec statistics gmac [network-interface] <enter | error
| rx | tx>
```

For example:

```
ACMEPACKET# show security ipsec statistics gmac M10
```

```
Host 0 Status:
```

Receive Statistics:

```
Total bytes           : 458422
Total frames          : 1802
Pause frames          : 0
Frames < 64 bytes     : 403
Frames 65-127 bytes   : 1053
Frames 128-255 bytes  : 5
Frames 256-511 bytes  : 0
Frames 512-1024 bytes : 4
Frames 1024-1500 bytes: 336
Frames > 1500 bytes   : 1
```

Transmit Statistics:

```
Total bytes           : 1180666
Total frames          : 8898
Good frames           : 8898
Pause frames          : 0
Frames < 64 bytes     : 438
Frames 65-127 bytes   : 7437
Frames 128-255 bytes  : 726
Frames 256-511 bytes  : 23
Frames 512-1024 bytes : 2
Frames 1024-1500 bytes: 272
Frames > 1500 bytes   : 0
```

Error Statistics:

```
Underflow errors      : 0
Collision errors      : 0
Excessive collisions  : 0
Carrier sense errors  : 0
Excessive deferrals   : 0
CRC errors            : 0
Alignment errors      : 0
< 64 bytes & CRC errors : 0
> 1518 bytes & CRC errors : 0
Length errors         : 0
FIFO overflows        : 0
Watchdog errors       : 0
```

show security status

The **show security status** command displays whether a particular interface on a session-director is IPsec enabled. If so, the hardware status of the security processor will be displayed. This command is entered as:

```
show security status [slot] [port]
```

For example:

```
ACMEPACKET# show security status 1 0
Interface '1, 0' IPSEC hardware status:
  Structure version      = 2
  Payload length (words) = 8
  Boot ROM code device ID = 0x0025
  Vendor ID              = 0x13a3
  Device ID              = 0x8450
  Hardware version       = 0x0000
  Subsystem type         = 0x00
  Software type          = 0x01
  Major software version = 0x01
  Minor software version = 0x02
  Memory config value    = 0x00000006
  System config register = 0x00a40020
  System clock 2 register = 0x00009555
  Saved parity status    = 0x0000
  Current parity status  = 0x0002
  POST test number       = 0x00
  POST sub-test number   = 0x00
  POST test result       = 0x00
  SW major revision      = 0x01
  SW minor revision      = 0x02
  SW maintenance revision = 0x00
  SW build number        = 0x09
```


Introduction

Transcoding is the ability to convert between media streams that are based upon disparate codecs. The Net-Net 9200 supports IP-to-IP transcoding for SIP, H.323, and IWF sessions, and can connect two voice streams that use different coding algorithms with one another.

This ability allows providers to:

- Handle the complexity of network connections and the range of media codecs with great flexibility
- Optimize bandwidth availability by enforcing the use of different compression codecs
- Normalize traffic in the core network to a single codec
- Enact interconnection agreements between peer VoIP networks to use approved codecs

By providing transcoding capabilities at the network edge rather than employing core network resources for the same functions, the Net-Net 9200 provides cost savings. It also provides a greater degree of flexibility and control over the codec(s) used in providers' networks and the network with which they interconnect.

In addition, placing the transcoding function in the SBC and at the network edge means that transcoding can be performed on the ingress and egress of the network. The Net-Net 9200 transcodes media flows between networks that use incompatible codecs, and avoids back-hauling traffic to a centralized location, alleviating the need for multimedia resource function processors (MRFPs) and media gateways (MGWs) to support large numbers of codecs. This maximizes channel density usage for the MRFPs and MGWs so that they can reserve them for their own specialized functions.

Hardware Support: the Net-Net 9200 TCUs and TCMS

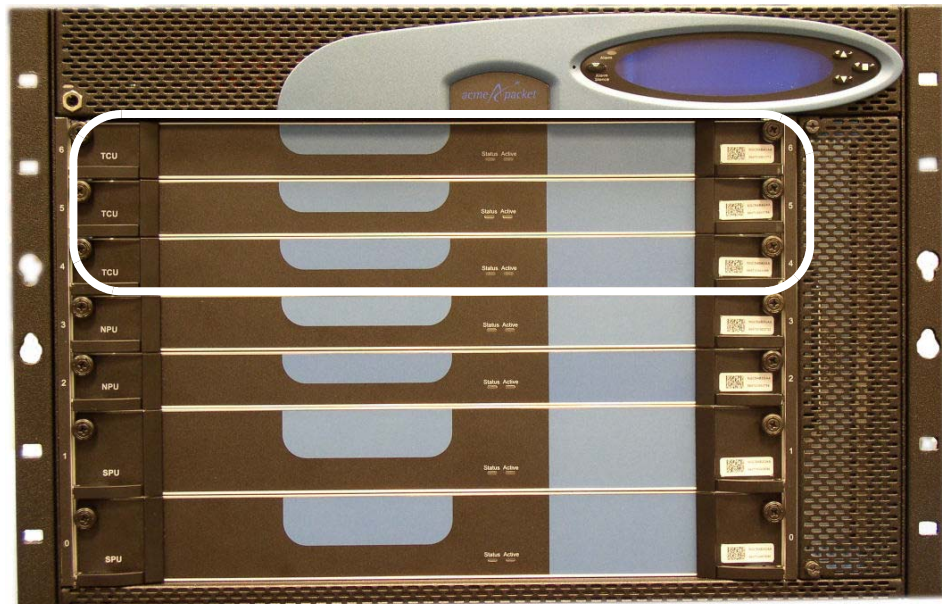
A Transcoding Unit (TCU) is the feature card that provides the DSP resources that enable the Net-Net 9200's transcoding feature. You can install one to three TCUs in a Net-Net 9200 to provide increasing transcoding capacity and redundancy. It is possible to deploy one TCU alone, two non-redundant TCUs, or deploy TCU cards in either of the following redundancy schemes:

- 1+1—Active-standby mode for high availability
- 2+1—Requires three TCUs, where one card is a hot-standby for the two active cards; if one of the active cards fails, the hot-standby takes over, making failures transparent for call and media processing

The TCUs are installed into the Net-Net 9200's top three feature card slots. For more information about the TCU hardware, and any procedures you might need in servicing the card, refer to the *Net-Net 9200 Hardware Guide*.

You do not need to obtain a license for this feature; transcoding is currently enabled automatically with your purchase of a TCU. As new the transcoding feature supports new codecs and functionality, licensing may be required.

The following picture shows you the location of the TCUs in the Net-Net 9200 chassis.



Each TCU may have one to four Transcoding Modules (TCM) installed. Media traffic marked for transcoding will flow through one TCM.

Note: All installed TCUs must be populated with the same number of TCMs.

Transcoding Capacity

Transcoding capacity depends on the following:

- Codecs used for transcoding
- Number of transcoding modules (TCMs) installed on the TCU; you can have between one and four TCMs per TCU
- Number of TCUs installed on the Net-Net 9200

The following table shows minimum session capacity (bi-directional IP-IP media session; one TCU equipped with one TCM) per codec pair with the following assumptions:

- 20ms packetization period or higher
- 100% use of the code pair identified by the corresponding column/row header
- As you reach the channel density limit, the session establishment rate will impact the gradient at which this limit will be reached, at high call rates there may be some sessions rejected before the limit is achieved
- EVRC-B is **NOT** licensed

Transcoding Capacity per TCM:

	G.711	G.726	G.729A/B	G.723	iLBC	G.728	GSM-FR	AMR-NB	EVRC	EVRCB	T.38
G.711	1000										
G.726	1000	1000									
G.729A/B	1000	890	760								
G.723	910	790	690	630							
iLBC	610	550	500	470	380						
G.728	500	460	420	400	330	290					
GSM-FR	870	760	660	610	460	390	590				
AMR-NB	800	710	620	580	440	380	560	530			
EVRC	740	650	580	540	420	360	520	500	470		
EVRCB	Not Licensed										
T.38	50										

Each TCM has a hard limit of 50 T.38 transcoding sessions. This is in addition being constrained by available processing resources that may be used by other transcoding tasks. Therefore a fully populated TCU can support 200 transcoded FAX calls, provided enough DSP resources are available.

Transcoding Capacity per TCM with EVRC-B licensed:

If EVRC-B is licensed on the system, there is an transcoding capacity impact for certain pairs (as highlighted).

	G.711	G.726	G.729A/B	G.723	iLBC	G.728	GSM-FR	AMR-NB	EVRC	EVRCB	T.38
G.711	1000										
G.726	1000	1000									
G.729A/B	1000	880	760								
G.723	910	790	680	630							
iLBC	610	550	500	470	370						
G.728	490	450	420	400	330	290					
GSM-FR	870	750	660	610	460	390	590				
AMR-NB	800	700	620	570	440	370	560	530			
EVRC	730	650	580	540	420	360	520	500	470		
EVRCB	530	480	440	420	340	300	410	390	380	320	
T.38	50										

For estimating specific transcoding hardware requirements to support your network deployment, please contact your System Engineers and sales team.

Transcoding Configuration

The Net-Net 9200 performs transcoding functions—allowing the entities with incompatible codecs to communicate with each other—between two call legs. The two endpoints can be located in one or two realms or networks. The Net-Net 9200 decides to transcode a call by evaluating messages in the SDP offer-answer transaction with respect to system configuration. An SDP offer can be in a SIP message such as an INVITE or a reINVITE, and contains information about the codecs the offerer would like to use. The answerer answers the SDP offer with its own set of supported codecs. reINVITEs are treated as new negotiations, with

respect to the actual SDP offerer and answerer. The Net-Net 9200 can manipulate an SDP message by reordering codec preference, and by adding and deleting codecs.

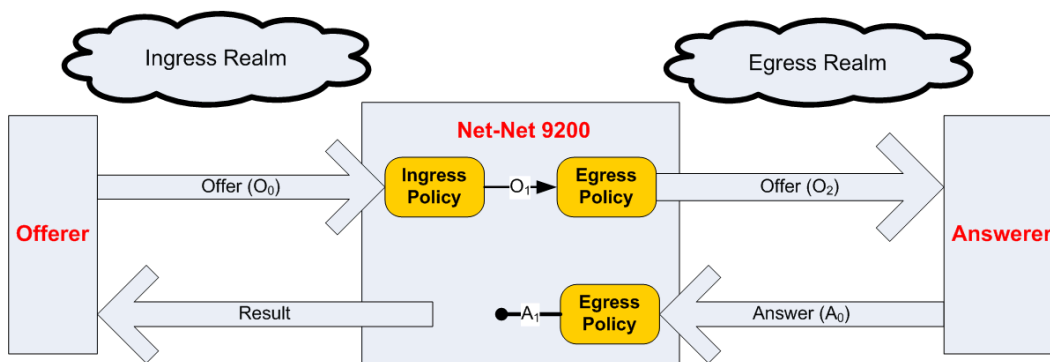
Transcoding Processing Overview

Transcoding processing is viewed in terms of the ingress and egress realms. The ingress realm is where the SDP offer is received by the Net-Net 9200. The egress realm is where the SDP offer is sent, and where the SDP answer is expected to be received from (i.e., the answerer's realm). A call is defined as transcodable if an egress or ingress policy exists for the session and if the session is not subject to media release, as specified in the realm configuration.

To understand the details of transcoding processing, refer to the following diagram. An SDP offer, O_0 , is received by the Net-Net 9200 in the ingress realm. The ingress transcoding policy is applied here and the SDP offer is transformed into O_1 . O_1 is then passed to and processed by the egress transcoding policy. This SDP message is then forwarded to the answerer as O_2 . The answerer replies with A_0 to the Net-Net 9200, which is subjected to the egress transcoding policy again and transformed into A_1 .

When policy dictates not to transcode the call, the Result SDP sent back to the offerer is based on the common codecs shared between A_1 and O_1 . The Net-Net 9200 first constructs the list of codecs that are present in both in O_1 and A_1 . Then, the Net-Net 9200 maintains the codec order from A_1 for the Result as it is sent to the offerer.

When policy dictates to transcode the call, the top transcodable codec in O_1 is used in the ingress realm and the top non-signaling codec in A_1 is used in the egress realm.



Defining Transcoding Policies

The following definitions are required for understanding transcoding processing:

DTMFable Codecs—Uncompressed codecs that are capable of properly transmitting a DTMF waveform. The only codecs designated as DTMFable are PCMU and PCMA.

FAXable Codecs—Uncompressed codecs that are capable of properly transmitting a T.30 waveform. The only codecs designated as FAXable are PCMU and PCMA.

Signaling Codecs—Non-audio codecs that are interleaved into a media stream but cannot be used on their own. The only codecs designated as Signaling Codecs by the Net-Net 9200 are telephone-event and CN (comfort noise).

Disabling an m= line—This is in reference to an m= line in an SDP message. It means setting the m= line's port to 0 (RFC 3264). Enabling an m= line means it has a non-zero port. The m= line's mode attribute (sendrecv/inactive/rtpcnonly, etc) is not considered.

A transcoding policy is created by configuring the following information:

- Which Codecs are allowed and which are denied in a realm.
- Which Codecs should be added to the SDP m= lines for an egress realm.
- The preferred order of codecs to indicate in an SDP m= line.
- The packetization time which should be enforced within a realm.

Ingress Policy

Incoming SDP is first subject to the ingress transcoding policy. If no transcoding policy is specified in the realm config for the ingress realm, or the m= lines in the SDP offer are disabled (by a 0 port number), the SDP is transformed to O₁ unchanged.

The ingress transcoding policy first removes all un-allowed codecs, as configured in the **allow** parameter (<codec>:no). For each un-allowed codec removed, its rtpmap and fmp attributes are also removed from SDP. Next, the remaining codecs are ordered with the **order** parameter. Ordering is when the transcoding policy rearranges the codecs in the SDP m= line. This is useful to suggest the codec preferences to impose within the egress realm. O₁ is then processed by the egress transcoding policy after a realm is chosen as the destination.

In practical terms, the ingress policy can be used for filtering high-bandwidth codecs from the access realm. It can also be used for creating a suggested, prioritized list of codecs to use in the ingress realm.

Egress Policy

The Net-Net 9200 applies egress transcoding policy to the SDP that has already been processed by ingress policy. The egress policy is applied before the SDP exits the system into the egress realm. If no egress transcoding policy is defined, or the SDP's m= lines are disabled (with a 0 port), the SDP is passed untouched from the ingress policy into the egress network.

The egress transcoding policy first removes all un-allowed codecs in the **allow** parameter (<codec>:no). Codecs on the **add-on-egress** list are not removed from the egress policy regardless of the how the **allow** parameter is configured. If the result does not contain any non-signaling codecs, the ptime attribute is removed from the SDP. Codecs not present in O₁ that are configured in the **add-on-egress** parameter are added to the SDP, only if O₁ contains one or more transcodable codecs.

Note: Transcoding can only occur for a call if you have configured the **add-on-egress** parameter in an egress transcoding policy.

If codecs with dynamic payload types (those between 96 and 127, inclusive) are added to the SDP, the lowest unused payload number in this range is used for the added codec.

The following rules are also applied for egress policy processing:

If O₁ contains at least one transcodable codec, the codecs listed in the Egress policy are added to the SDP.

- telephone-event, as configured in add-on-egress will only be added if O_1 contains at least one DTMFable codec.
- T.38, as configured in add-on-egress will only be added if there is no T.38 and there is at least one FAXable codec (G711Fall Back (FB)) in O_1 . T.38 will then be added as a new m= image line to the end of SDP.

If G711FB is not allowed in the egress policy, the Net-Net 9200 disables the m= line with the FAXable codec. Otherwise if G711FB is allowed, pass it through the regular offer processing allowing/adding only FAXable codecs.

- G711FB, as configured in add-on-egress will only be added if there is no G711FB and there is T.38 in O_1 . G711FB will then be added as a new m= audio line to the end of SDP.

If T.38 is not allowed in the egress policy, the Net-Net 9200 disables the m= image line. Otherwise if T.38 is allowed, pass it through the regular offer processing.

If the result of the egress policy does not contain any non-signaling codecs, audio or video, the m= line is disabled, by setting the port number to 0.

The m= line is next ordered according to rules for the order parameter.

Finally, all attributes, a= lines,ptime attribute, and all other unrecognized attributes are maintained from O_1 . Likewise, appropriate attributes for codecs added by the add on egress parameter are added to SDP. Finally, rtpmap and fmtp parameters are retained for codecs not removed from the original offer. The result of all this is O_2 , as shown in the overview diagram.

In practical terms, transcoding policies can be used to normalize codecs and ptime in the core realm where the network conditions are clearly defined.

Transcoding policies can also be used to force the most bandwidth-conserving codecs anywhere in the network.

Post Processing

If any errors are encountered during the Ingress and Egress policy application, or other violations of RFC3264 occur, the call is rejected. If O_2 does not contain any enabled m= lines at the conclusion of the initial call setup, the call is rejected. If O_2 does not contain any enabled m= lines at the conclusion of a reINVITE, the reINVITE is rejected and the call reverts back to its previous state.

Transcoding Policy Definition

Transcoding policies describe how to manipulate SDP messages as they cross the Net-Net 9200. The Net-Net 9200 bases its decision to transcode a call on transcoding policy configuration and the SDP. Each transcoding policy specifies a set of rules to be used for determining what codecs are retained, removed, and how they are ordered within SDP.

Syntax

The following parameters are used to create a transcoding policy. Their syntax is described inline.

allow

allow—The allow parameter configures the codecs that are allowed and/or removed from the SDP. A blank list allows nothing, * allows all codecs, **none** removes all codecs, the **:no** designation blocks the specific codec or class of media, and the **:force** designation is used to remove all non-forced codecs.

The **allow** parameter is configured in the following way:

- `<codec>:no`—blocks the specific codec
- `*`—allow all codecs.
- `<codec>:force`—If any forced codec is present in an SDP offer, all non-forced codecs are stripped from the m- line.
- **audio:no**—audio m= line is disabled
- **video:no**—video m= line is disabled

For example, if you configure PCMU in the allow parameter, the PCMU codec, received in an SDP message is allowed on to the next step of transcoding processing, and all other codecs are removed.

The order of precedence is for removing codecs according to transcoding policy is:

1. `<codec>:no`—Overrides all other allow parameter actions.
2. **audio:no** / **video:no**. An allow line like "allow PCMU audio:no" disables the PCMU m= line because "audio:no" has a higher precedence than the specific codec.
3. `<codec>:force`
4. `<codec>` Specific codec name and those codecs configured in the add-on-egress list.
5. `*` has the lowest precedence of all flags. For example "**allow * PCMU:no**" allows all codecs except PCMU.

order

order—The order parameter is used to re-order the codecs in the m= line as the SDP is passed on to the next step. This parameter overwrites the order modified by the **add-on-egress** command, when relevant. The following is valid syntax for this parameter:

- `<blank>`—Do not re-order codecs
- `*`—You can add a `<codec>` before or after the `*` which means to place all unnamed codecs before or after (the position of the `*`) the named codec. For example:
- `<codec> *`—Puts the named codec at the front of the codec list.
- `* <codec>`—Puts the named codec at the end of the codec list.
- `<codec1> > * <codec2>`—Puts `<codec1>` first, `<codec2>` last, and all other unspecified codecs in between them.
- `<codec>`—When the `*` is not specified, it is assumed to be at the end.

Any codec name is allowed in the **order** parameter, even those not defined or not transcodable. An `*` tells the order parameter where to place unspecified codecs with respect to the named codecs. Refer to the examples below.

- <blank>—do not reorder m= line
- PCMU *—Place PCMU codec first, all others follow
- * PCMU—Place PCMU codec last, all others proceed PCMU
- G729 * PCMU—Place G729 codec first, PCMU codec last, all others remain in between
- PCMU—If * is not specified, it is assumed to be at the (PCMU *).

Add on Egress

add-on-egress—This parameter adds a codec to the SDP's m= line only when the transcoding policy is referenced from an egress realm (except in one 2833 scenario). Codecs entered for this parameter are added to the front of the m= line. Signaling codecs are added to the end of the m= line.

Transcoding can only occur if this parameter is configured. There is a special case for 2833 support where the add-on-egress parameter is configured for an ingress realm. See [RFC 2833 Scenario 2 \(774\)](#) for details.

Packetization Time

packetization-time—This parameter specifies a media packetization time in ms to use within the realm referencing this transcoding policy. Packetization time You must also enable the force ptime parameter to enable transrating in conjunction with configuring the packetization time. See the [Transrating \(781\)](#) section for more information.

Answer Processing and Examples

Unoffered Codec Reordering

According to RFC 3264, the answerer can add codecs that were not offered to the Answer. The answerer may add new codecs as a means of advertising capabilities. RFC 3264 stipulates that these unoffered codecs must not be used. The RFC does not dictate where in the m= line these codecs can appear and it is valid that they may appear as the most preferred codecs.

In order to simplify the answer processing, the Net-Net 9200 moves all unoffered codecs in A₀ to the back of the SDP answer before any other answer processing is applied.

Non-transcoded Call

The decision to transcode is based on the top non-signaling codec in A₁. If the top A₁ codec is present in O₁, the call proceeds, non-transcoded. This is the rule for non-signaling codecs (i.e., not RFC 2833 nor FAX).

Transcoded Call

The following two conditions must then be met to transcode the call's non-signaling media:

- The top A₁ codec is not present in the O₁ m= line
- The top A₁ codec was added by the egress policy

If these rule are met, the Net-Net 9200 will transcode between the top A₁ codec and the top transcodable, non-signaling O₁ codec.

Voice Transcoding

The following examples use the ingress and egress transcoding policies listed at the top of each scenario. The examples use changing SDP offers and answers, which contribute to unique results, per example. The effects of the SDP offers and answers are explained in each example.

Voice Scenario 1

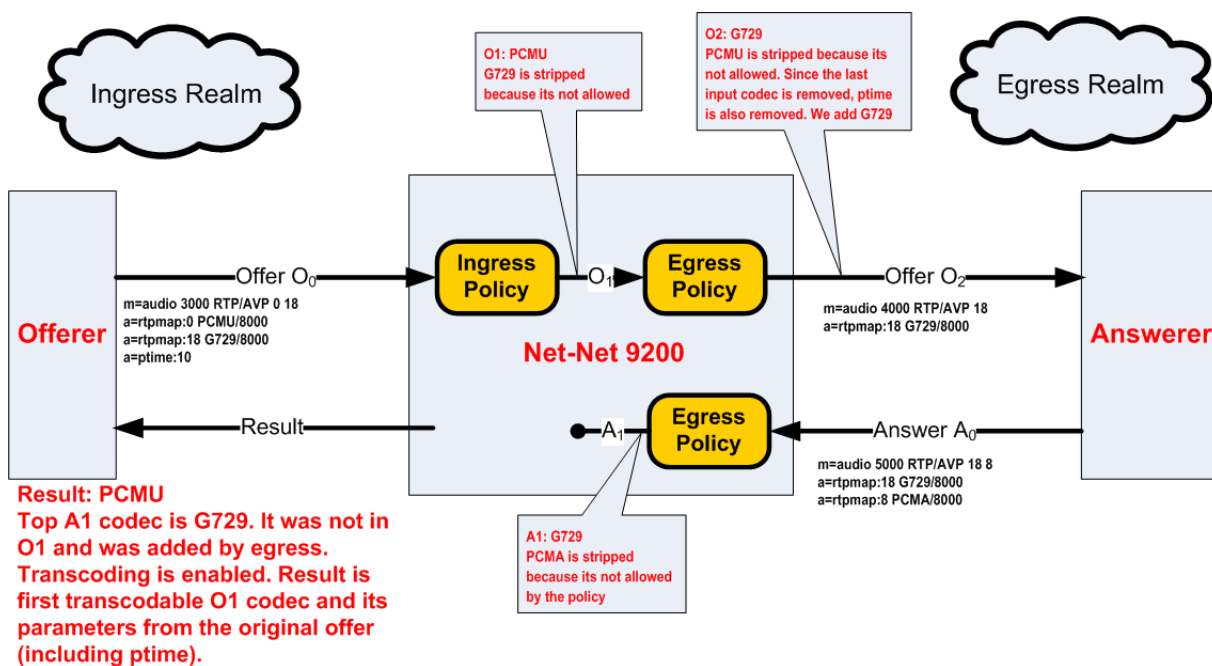
The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow	PCMU GSM	allow	G729 GSM G722
add-on-egress	PCMU	add-on-egress	G729
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

Note: The codec in the ingress policy's add-on-egress parameter has no effect in the following examples. Its presence would have an effect if a reINVITE was initiated from egress realm, effectively reversing the roles of the transcoding policies.

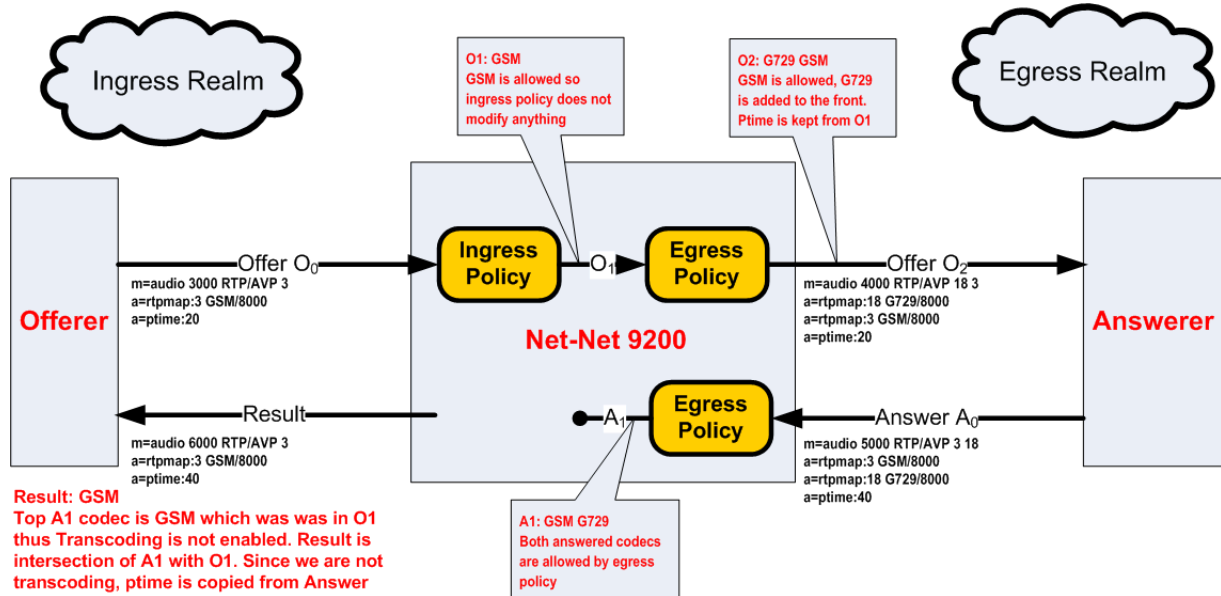
1. In the following diagram, PCMU and G729 are offered. Ingress policy removes G729 and allows PCMU. The egress policy adds G729 and strips PCMU from offered SDP and forwards it on to the answerer (ptime is also removed because the last codec was removed).

The SDP answer agreed to use G729 and adds PCMA. The egress policy then strips PCMA from the SDP answer. At this point, the top codec in A₁, G729 is checked against O₁. Since G729 is not present in O₁, it is transcoded to PCMU.

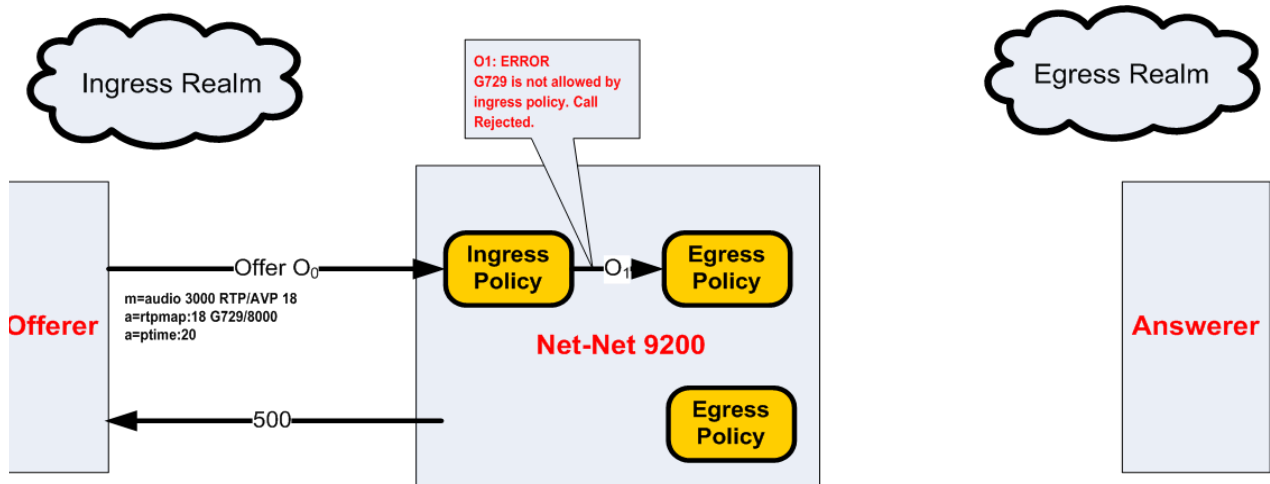


- In the following diagram, GSM is in the original SDP offer. It is then passed through to O_1 . Egress policy adds G729 and retains ptime from GSM and sends this to the answerer as O_2 .

The SDP answer agrees to use G729 and GSM, but prioritizes GSM. The egress policy allows both codecs through, unchanged. Because A1 and O1 both have GSM, it is used for the non-transcoded call. Ptime is copied from A_0 to the result.

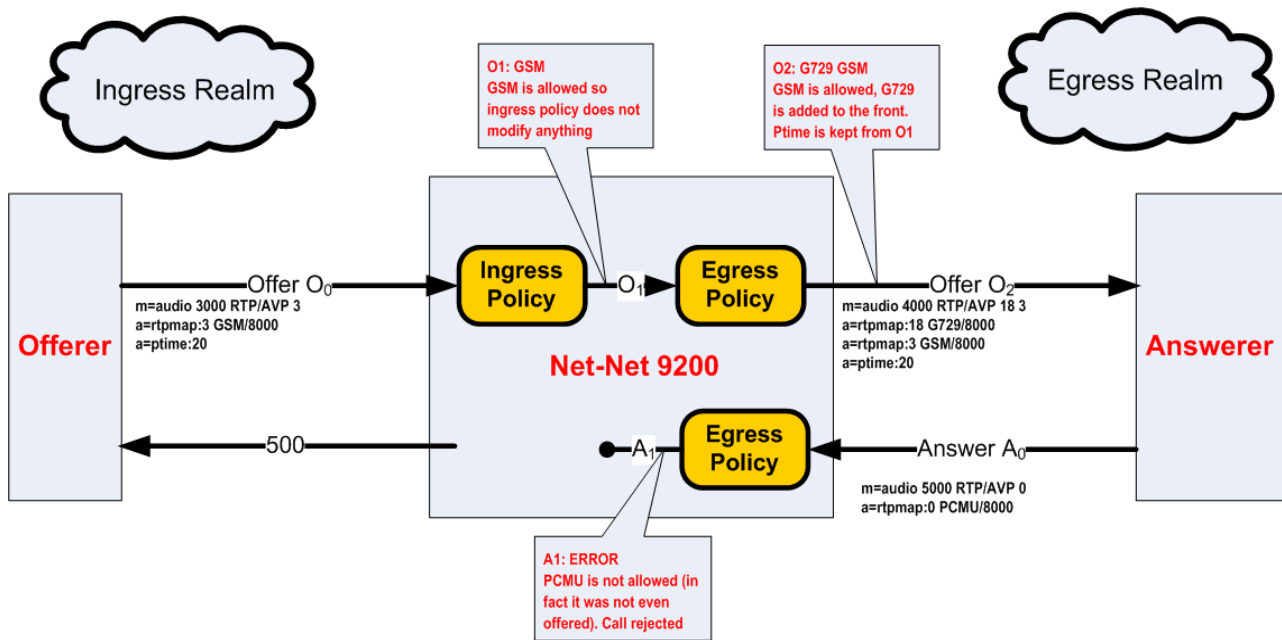


- In the following diagram, G729 in the original SDP offer. Because once G729 is removed, no non-signaling are left in O_1 , thus the call is rejected.



4. In the following diagram, GSM is in the original SDP offer. It is then passed through to O_1 . Egress policy adds G729 and retains ptime from O_1 and sends this to the answerer as O_2 .

The SDP answer states that the answerer wants to use PCMU. This is a violation of the RFC3264. Therefore the call is rejected.

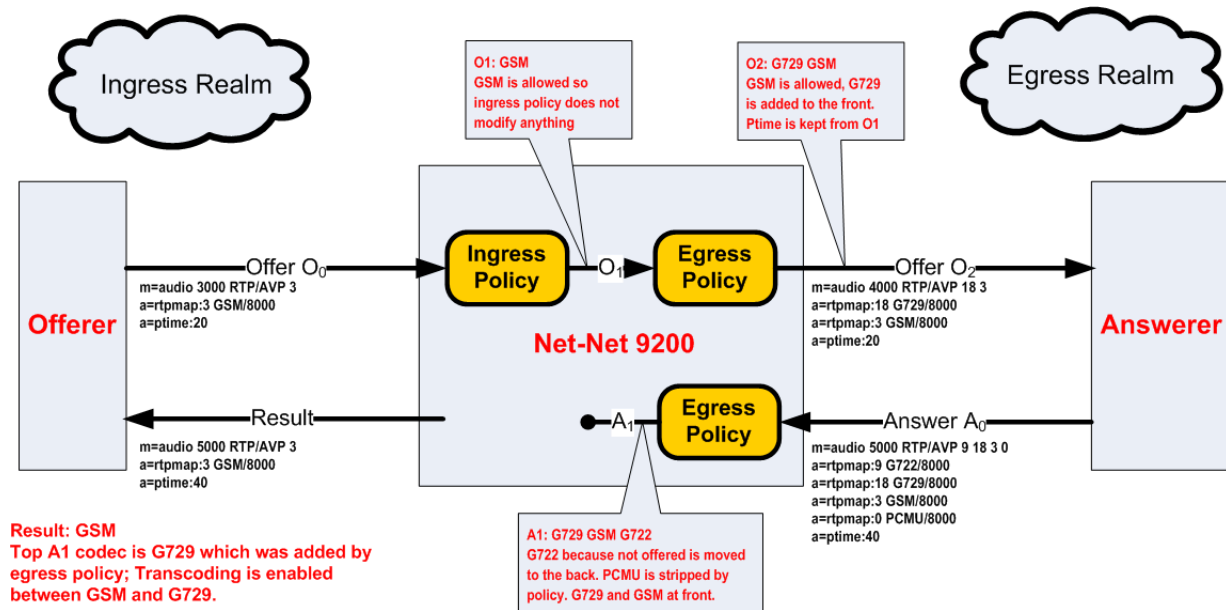


In this example, when the negotiation fails, the Net-Net 9200 sends a 500 message to the offerer and a BYE message to the answerer.

5. In the following diagram, GSM is in the original SDP offer. It is then passed through to O_1 . Egress policy adds G729 and retains ptime from O_1 and sends this to the answerer as O_2 .

The SDP answer replies with G722 G729 GSM and PCMU. PCMU is stripped by policy, G722 is moved to the back of the answer because it was not offered.

The top A_1 codec was not in O_1 , and was added by egress policy, therefore the call is transcoded between GSM and G729.

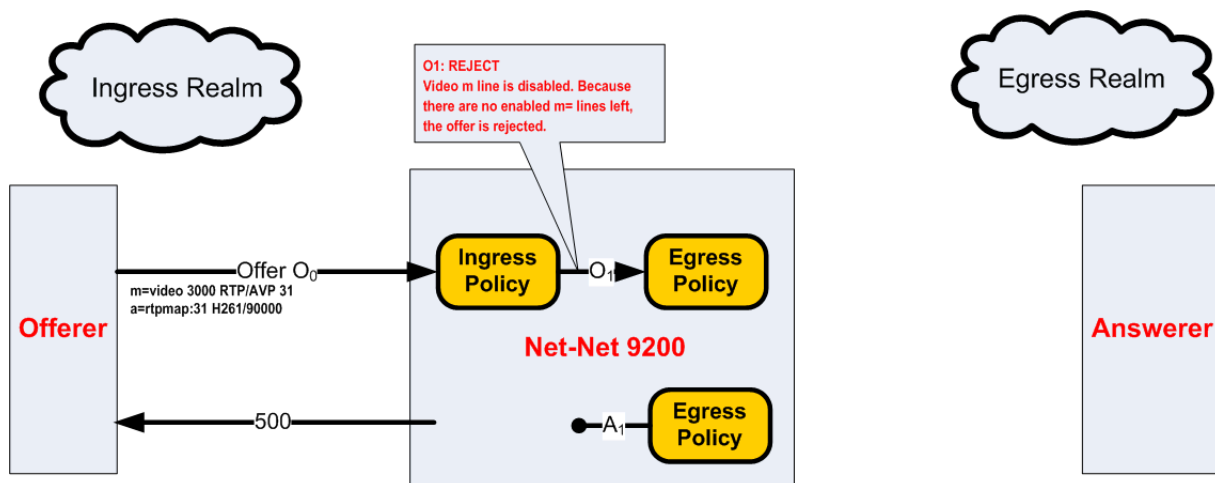


Voice Scenario 2

The following ingress and egress policies are used for scenario 2.

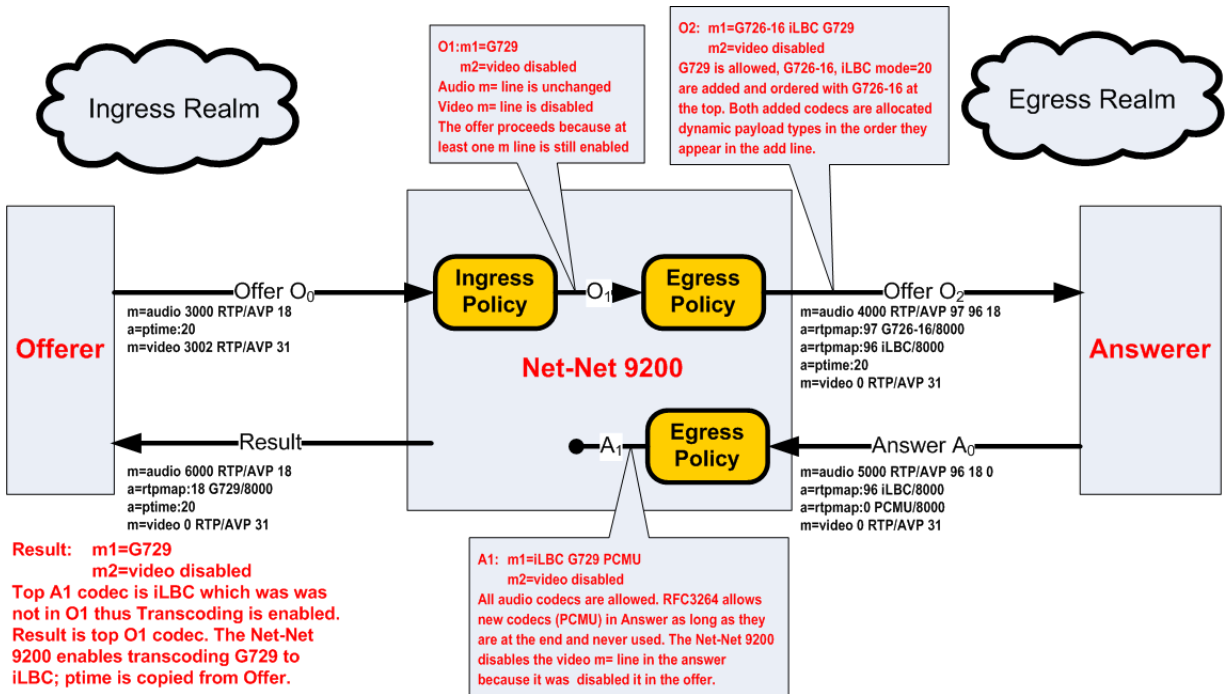
Ingress Policy		Egress Policy	
allow	Video:no PCMU:force * PCMA:force	allow	* PCMA:no
add-on-egress		add-on-egress	iLBC G726-16
order		order	G726-16 * PCMU
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, a video m= line is offered. The ingress policy disables the video m= lines. With no enabled m= lines left, the call is rejected.



2. In the following diagram, G729 and video are offered to the Net-Net 9200. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order parameter. The ptime is maintained between O_0 and O_2 . Both added codecs are allocated dynamic payload types in the order they appear in their m= line. A disabled Video m= line is passed on to the answerer.

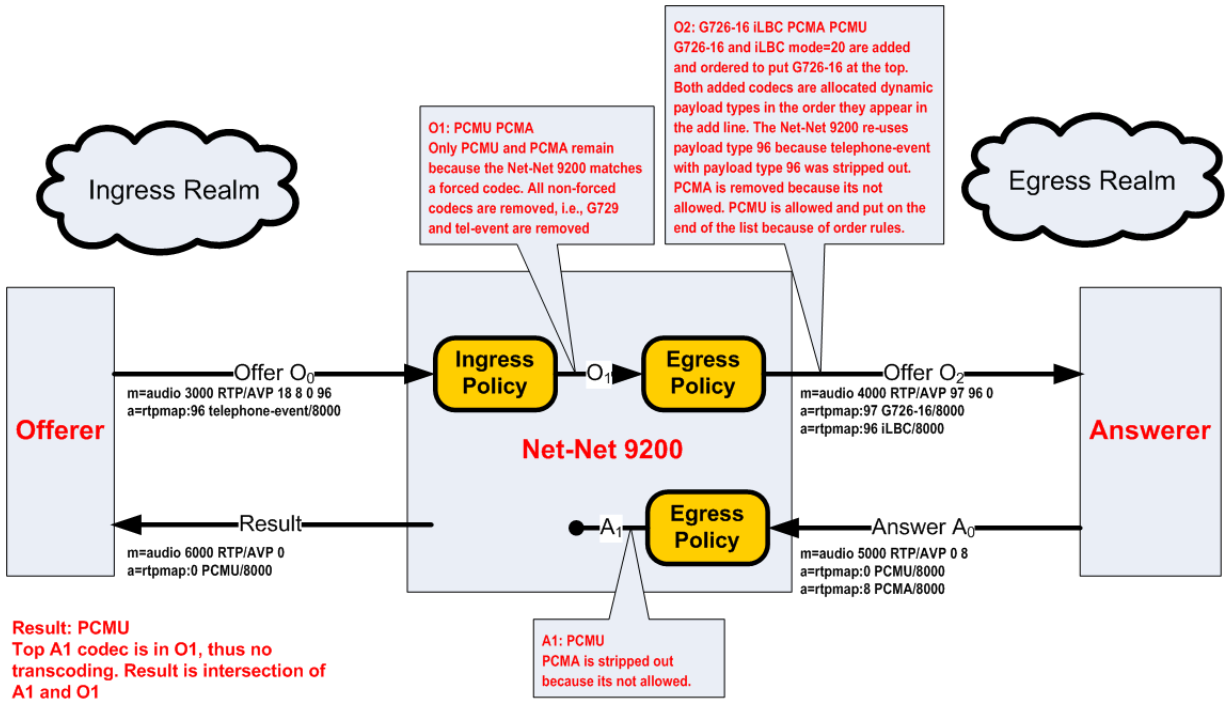
The SDP answer agreed to use iLBC, G729, and adds PCMU, and reorders them as stated. The disabled video m= line is maintained. At this point, the top codec in A_1 , iLBC is used and transcoded with the top codec in O_1 , G729.



3. In the following diagram, G729 and video are offered to the Net-Net 9200. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order parameter. The ptime is maintained between O_0 and O_2 . Both added codecs are allocated dynamic payload types in the order they appear in their m= line.

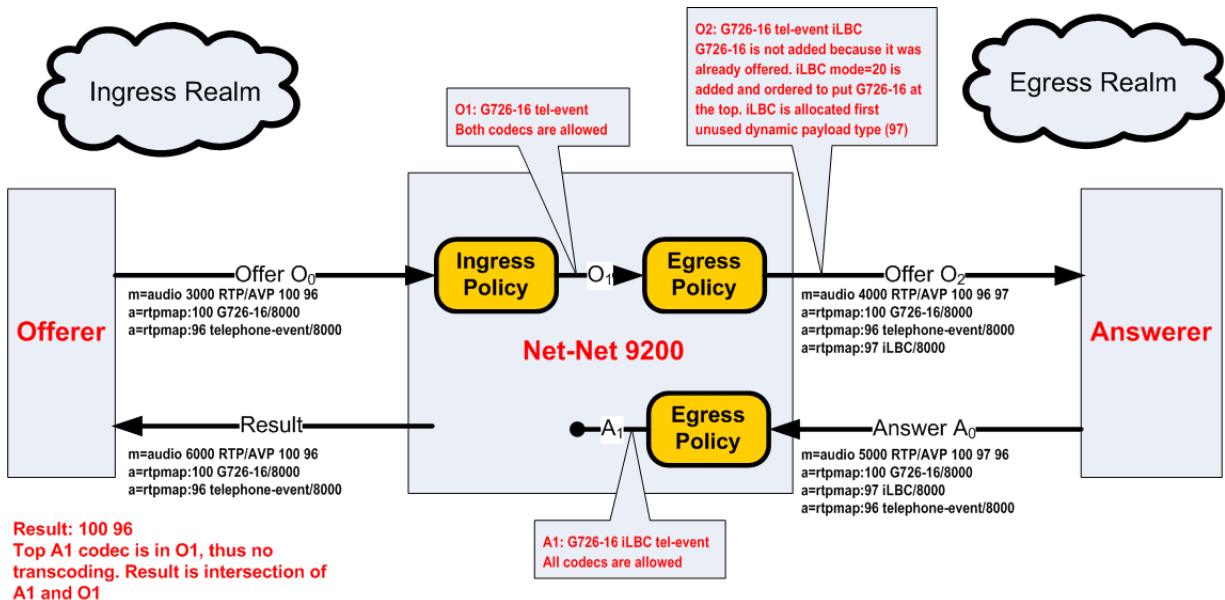
The SDP answer only wants to use PCMU and PCMA. The egress policy removes PCMA and passes only PCMU to the offerer. Because PCMU was in O_1

and is now the only codec in A_1 , it is used, and no transcoding is used between the endpoints.



- In the following diagram, G726-16 and telephone-event are offered to the Net-Net 9200. Ingress policy allows both codecs. The egress policy adds iLBC, and then orders the codecs according to the order parameter.

The SDP answer agreed to use all codecs, but reorders them with G726-16 in the top position. Because G726-16 is the top codec in A_1 , and it is also present in O_1 , it is used for this call without any transcoding.

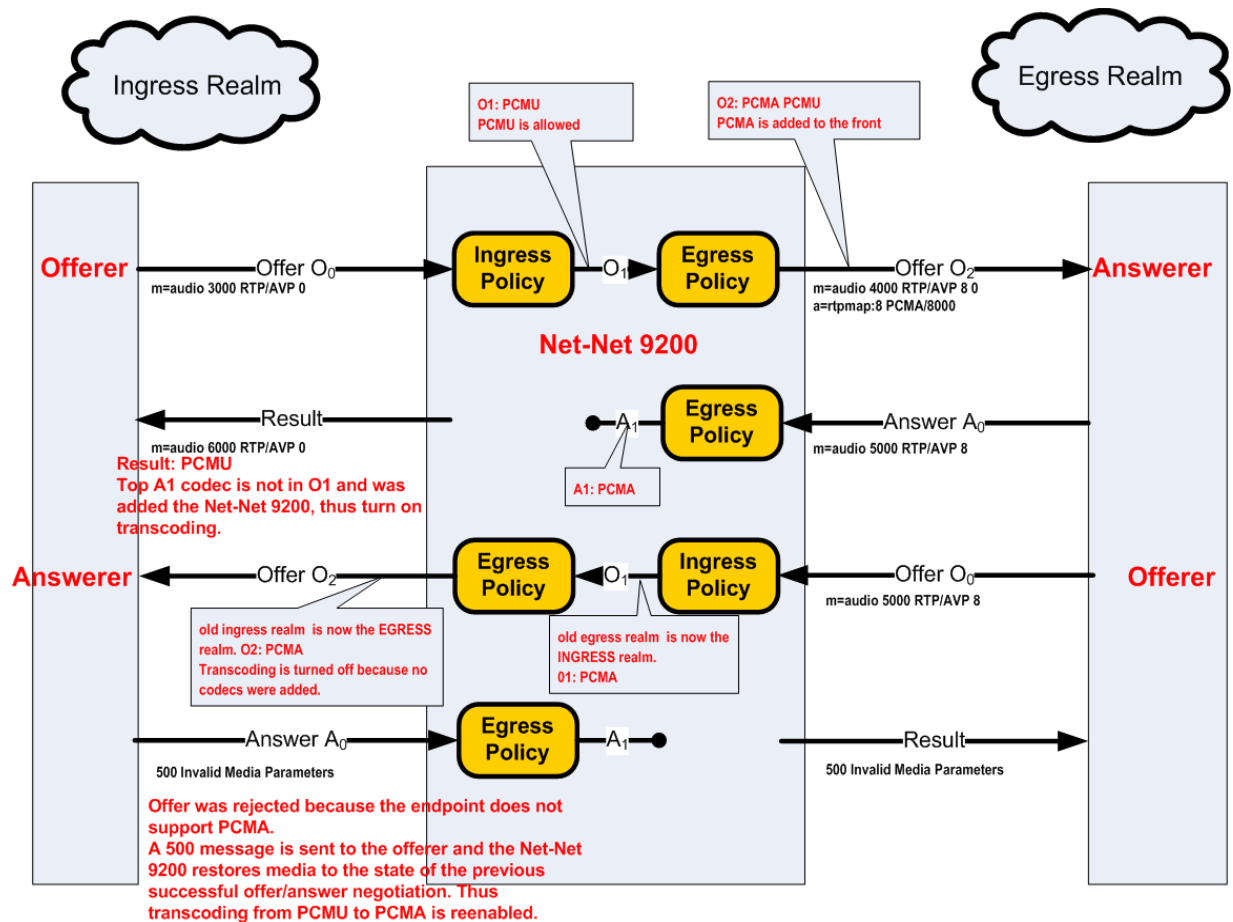


Voice Scenario 3

Voice scenario 3 involves reINVITEs. The following ingress and egress policies are used for scenario 3.

Ingress Policy		Egress Policy	
allow	PCMU G729	allow	*
add-on-egress		add-on-egress	PCMA
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, the answerer sends a reINVITE after a previous transcoding session was established. The original offerer and answerer swap roles. The new offerer rejects the SDP offer and the call reverts to the state negotiated in the original SDP negotiation.

**RFC 2833 Transcoding**

RFC 2833 defines an RTP payload that functions interchangeably with DTMF Digits, Telephony Tones and Telephony Signals. The Net-Net 9200 can monitor audio stream for in-band DTMF tones and then can convert them to data-based

telephone-events, as sent in RFC2833 packets. This section explains how the Net-Net 9200 transcodes between these RTP-based telephone events and in-band DTMF tones carried by G711. DTMF tones can only be transported in non compressed codecs. The Net-Net 9200 supports two DTMFable non-compressed codecs: PCMU (G711 μ) and PCMA (G711A).

Note: The following line is added to SDP whenever telephone-event is added on egress: `a=fmtp:101 0-15`

The following two scenarios describe when telephone-event to DTMF transcoding takes place:

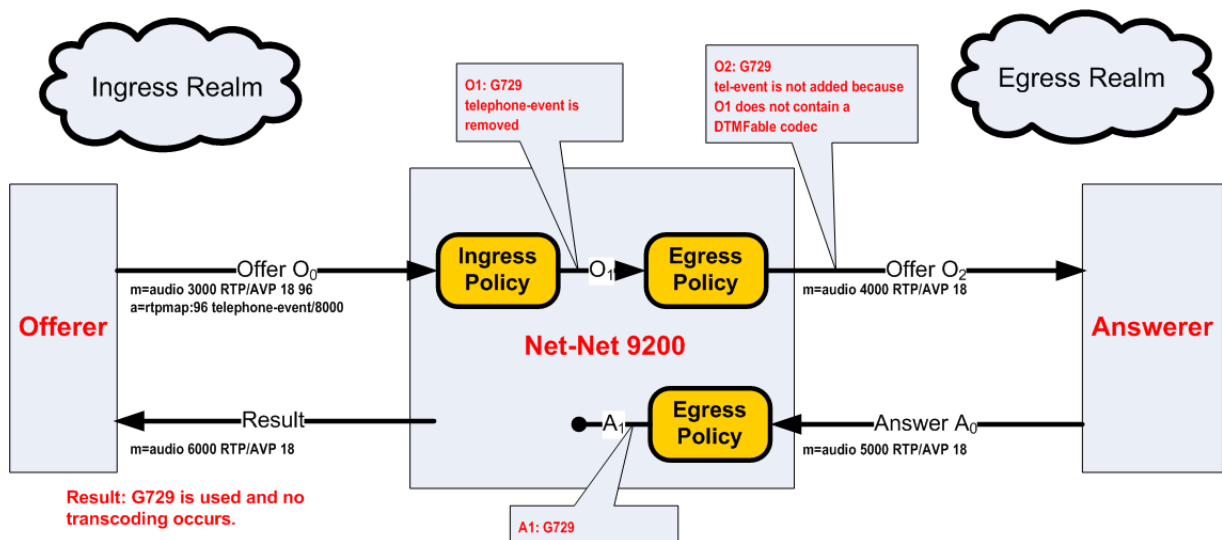
RFC 2833 Scenario 1

The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow	* telephone-event:no	allow	* PCMA:no
add-on-egress		add-on-egress	telephone-event
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

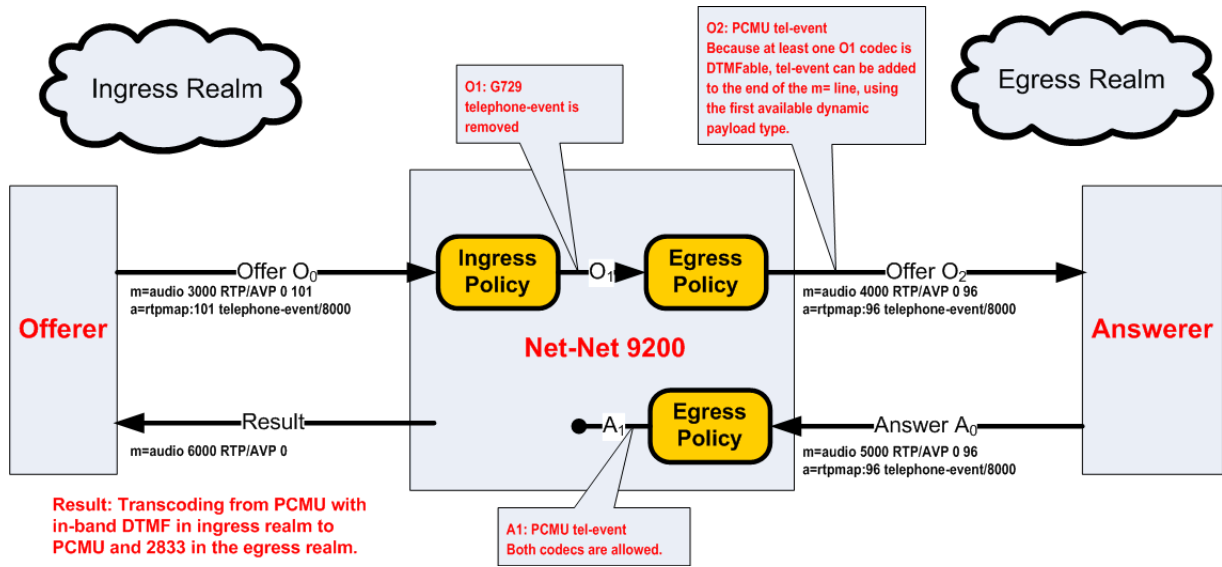
1. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was not added by the egress policy because the remaining audio codec in O_1 was not DTMFable. G729 was the only codec forwarded on to the answerer.

The SDP answer agreed to use the remaining audio codec, G729. A_0 is unaltered by egress policy, and forwarded as the Result to the offerer. Therefore, G729 is used in both the ingress and egress realms, the call does not support RFC 2833, and the call is not transcoded.



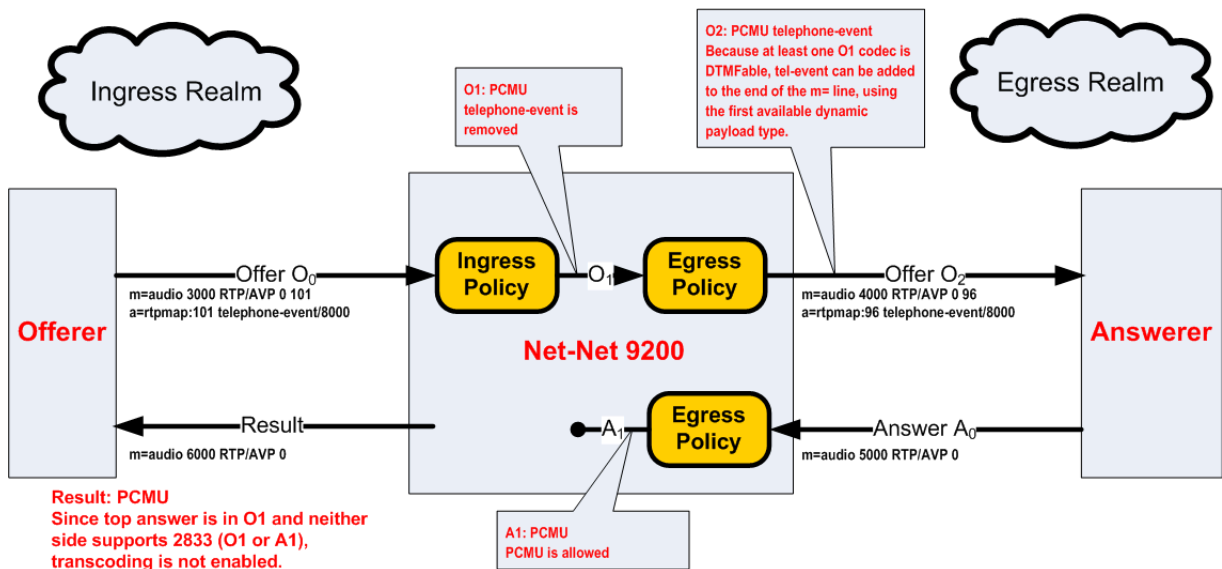
2. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was added by the egress policy because the remaining audio codec in O_1 was DTMFable. PCMU and

telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.



This case illustrates when the answerer supports audio and RFC 2833, but the offerer supports audio with inband DTMF. The Net-Net 9200 transcodes between RFC2833 in the egress realm to in-band DTMF on the ingress realm.

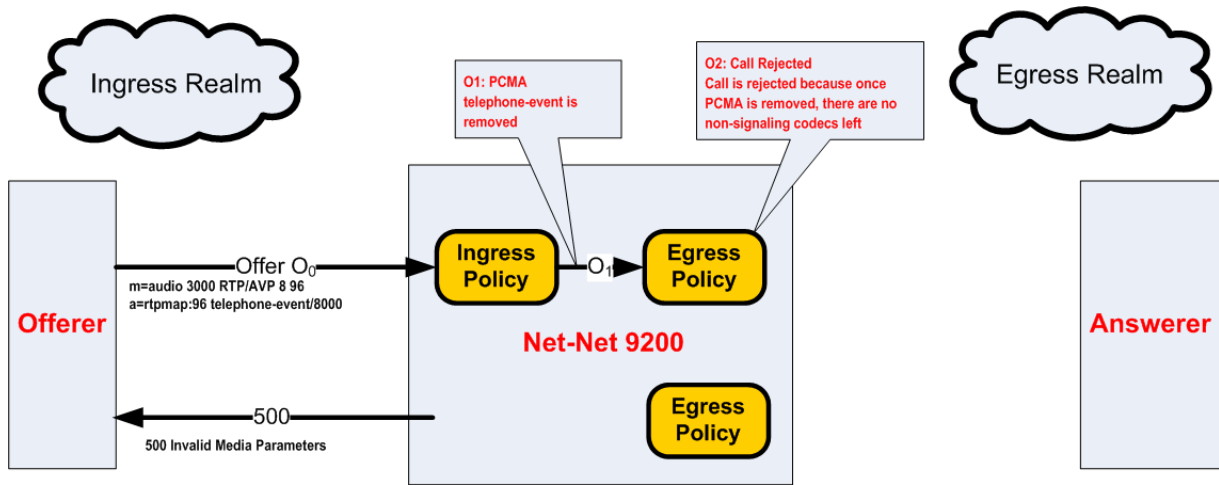
3. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event is added by the egress policy because the remaining audio codec in O_1 was DTMFable. PCMU and telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.



The SDP answer only agreed to use PCMU. When A_0 reaches the egress policy, it is passed along through the Net-Net 9200 to the offerer. Because telephone-

event was not answered by the answerer and not supported in O_1 , it can't be used. Transcoding is therefore not used for this call.

4. In the following diagram, telephone event was offered by the offerer and was stripped by ingress policy. Since PCMA was also stripped by the egress policy, leaving no non-signaling codecs, the call is rejected. A 500 message is sent back to the offerer.

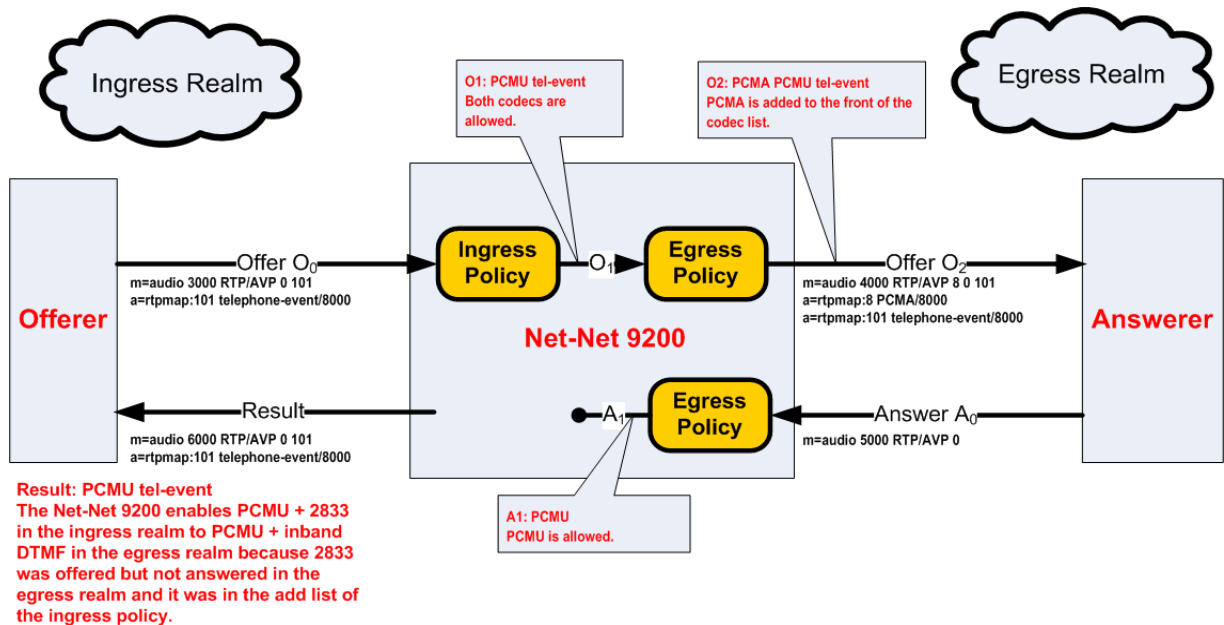


RFC 2833 Scenario 2

The following ingress and egress policies are used for RFC2833 scenario 2.

Ingress Policy		Egress Policy	
allow	*	allow	*
add-on-egress	telephone-event	add-on-egress	PCMU
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O_1 , and PCMA is added as it is sent to the answerer. The SDP answer, A_0 disables all codecs but PCMU.

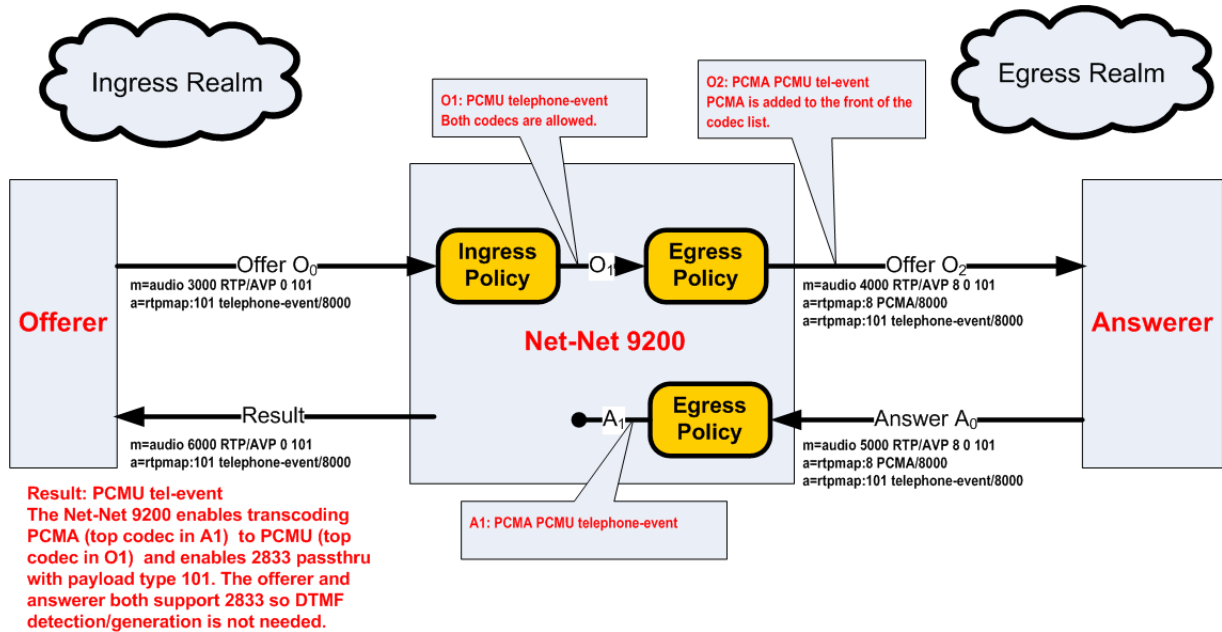


The Net-Net 9200 adds telephone-event to the result because it is listed on the ingress policy's add-on-egress parameter and present in the offerer's SDP.

Note: This is the only time the add list of an ingress policy is utilized as a check.

The result SDP includes PCMU and telephone-event in the ingress realm, which is transcoded to PCMU with in-band DTMF in the egress realm.

2. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O_1 , and PCMA is added as it is sent to the answerer. The SDP answer supports all three codecs offered, with PCMA added on top.



The answerer responds with PCMA as the preferred codec in A_0 . The Net-Net 9200 compares A_1 to O_1 to make the transcoding decision. PCMA is the top codec in A_1 and is transcoded to PCMU, the top codec in O_1 . Also, because telephone-event is supported by both sides of the call, it is passed through without any transcoding necessary.

This case illustrates when both endpoints are capable of sending and receiving telephone-event. Regardless of whether the audio portion of the call is transcoded, the telephone-event messages are passed through the system untouched, thus not requiring transcoding resources. This is known as telephone-event pass-through.

FAX Transcoding

FAXes are transmitted in a call as either T.30 and T.38 media. T.30 FAX is binary in-band media carried over G.711. The Net-Net 9200 can transcode between T.38 and a faxable codec. The supported faxable codecs are PCMU and PCMA.

T.30 can only be transported in non-compressed codecs. The two non-compressed codecs supported by the Net-Net 9200 are PCMU (G711 μ) and PCMA (G711A). If a transcoding realm does not support an uncompressed codec, T.30 can not be supported in that realm. Alternatively, G711FB may be allowed specifically for FAX only.

The Net-Net 9200 uses an internal codec called G711FB (G711 - Fall Back) that is an umbrella codec of all FAXable codecs. G711FB will default to PCMU for the purpose of offering a faxable codec. You can remap G711FB to PCMA by configuring the media-profile for it appropriately. G711FB's only use is for FAX transcoding.

FAX transcoding is triggered when you configure the add on egress parameter with either T.38 or G711FB. In a FAX scenario, when the transcoding policy adds either T.38 or G711FB, a new m= line is added to the SDP. When adding T.38, the new m= line specifies the T.38 codec. When adding G711FB, the new m= line specifies PCMU (or alternatively PCMA).

Once added, m= lines can not be deleted in the context of a call. The Net-Net 9200 maintain all m= lines between itself and an endpoint throughout the course of call. All m= lines not in use can be disabled by setting their receive port to 0, but they can not be removed from the SDP.

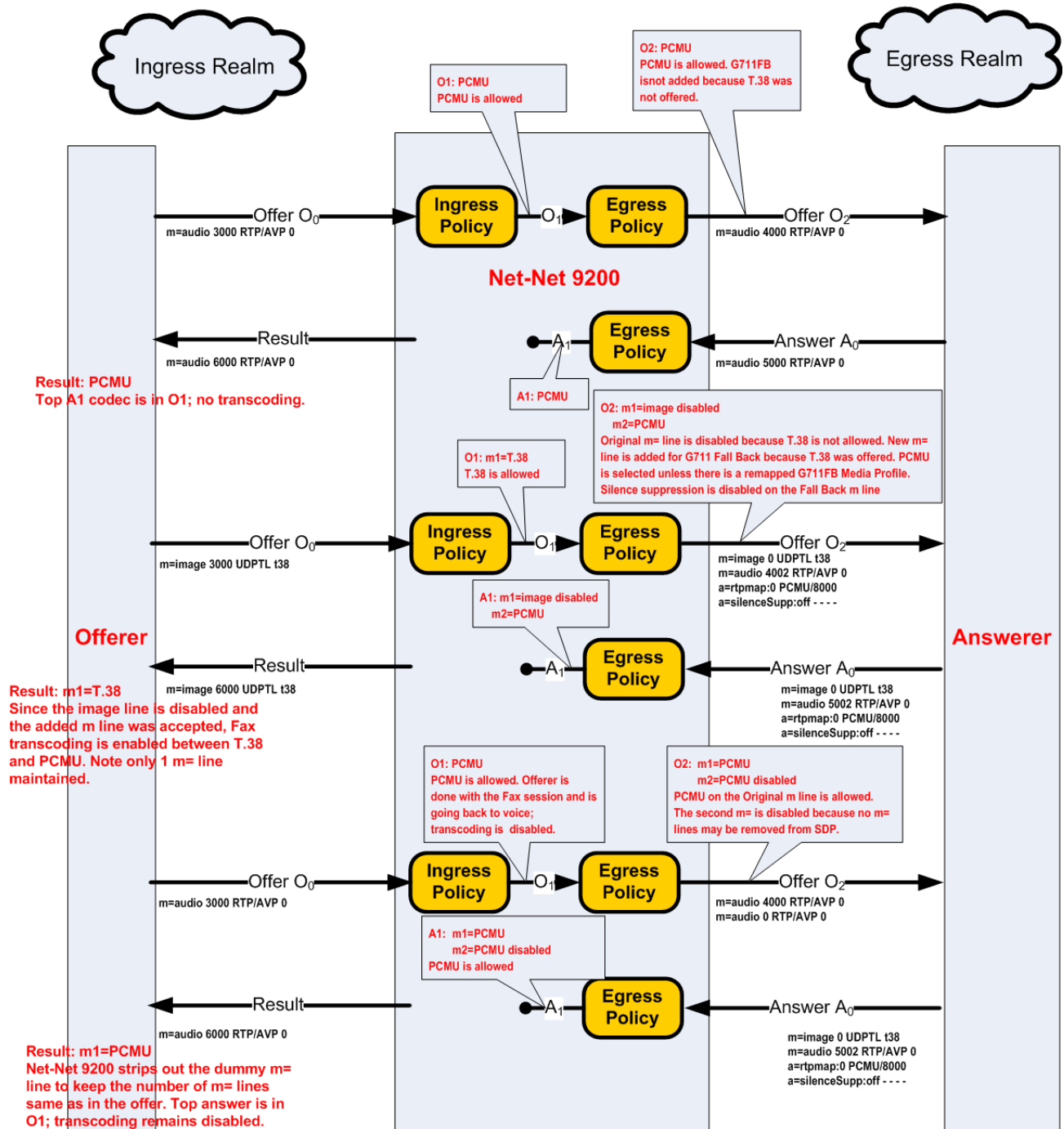
FAX Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow	*	allow	T.38:no
add-on-egress		add-on-egress	G711FB
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, there are three offer-answer exchanges. Initially a PCMU-to-PCMU session is negotiated. Next, a T.38 to PCMU session is

negotiated. Finally, the session reverts to non-transcoded PCMU to PCMU state.



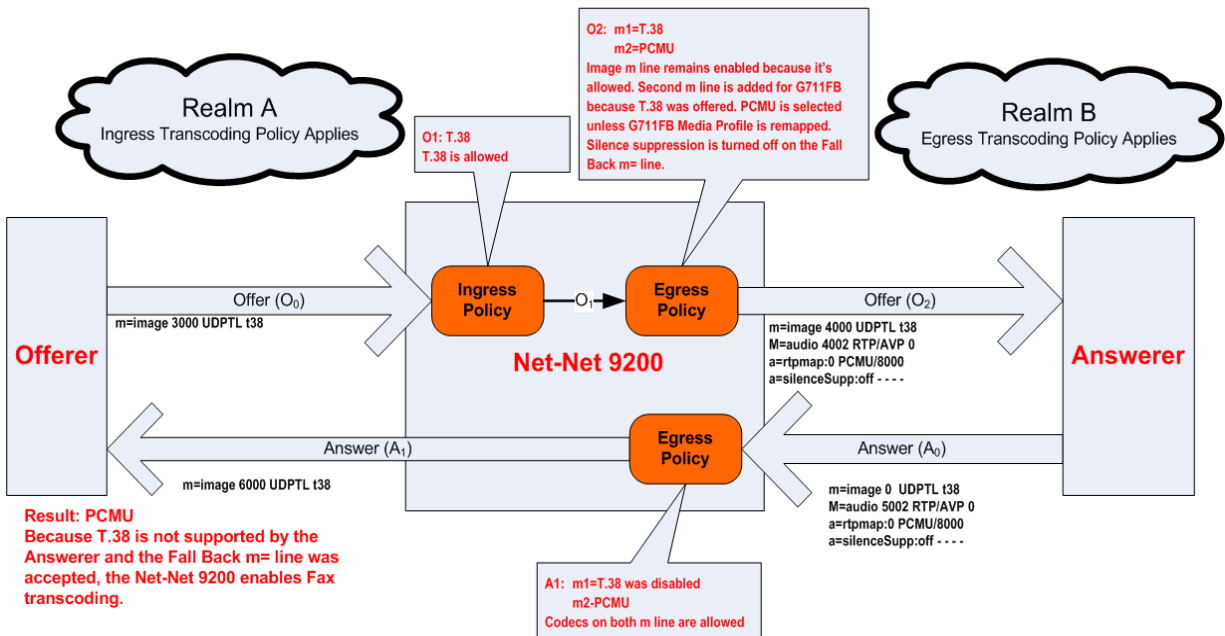
FAX Scenario 2

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow	*	allow	*
add-on-egress		add-on-egress	G711FB
order		order	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

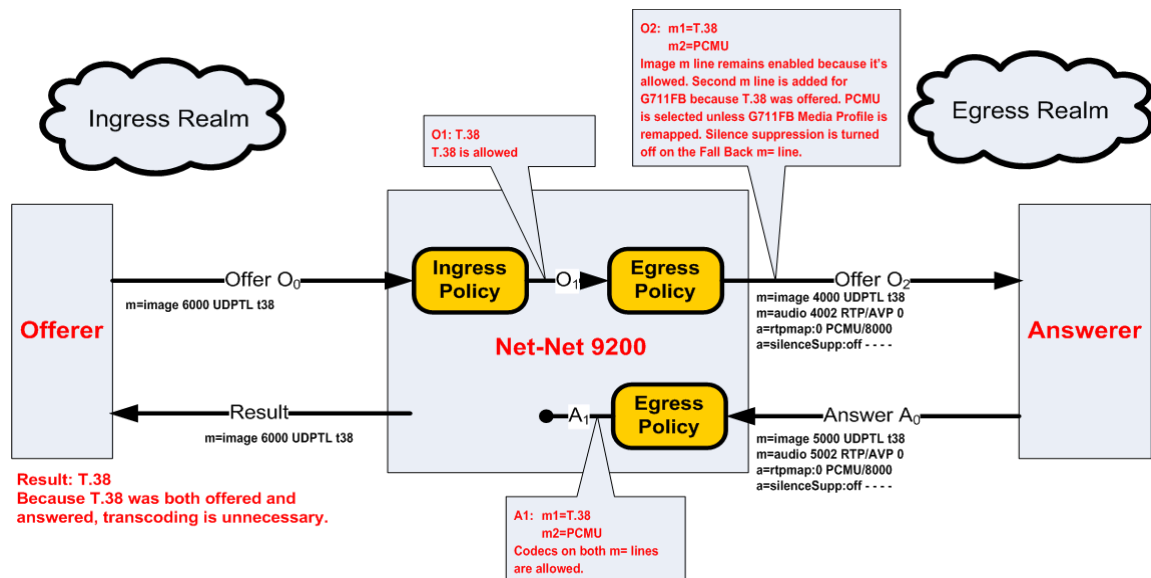
- In the following diagram, T.38 is offered to the Net-Net 9200. A second m= line was added to O₁ that included a G711FB codec (PCMU).

The SDP answer agreed to PCMU, but disabled T.38. When the Net-Net 9200 forwarded the SDP in A₁ to the answerer, it stripped the second m= line. Because A₁ rejects T.38 m= line, but accepts the PCMU m= line, FAX transcoding is enabled.



- In the following diagram, T.38 is offered to the Net-Net 9200. A second m= line was added to O₁ that included a G711FB codec (PCMU).

The SDP answer agreed to PCMU and T.38. Because both O_1 and A_1 support T.38, the call proceeds without transcoding.



FAX Scenario 3 - Special 200OK Case

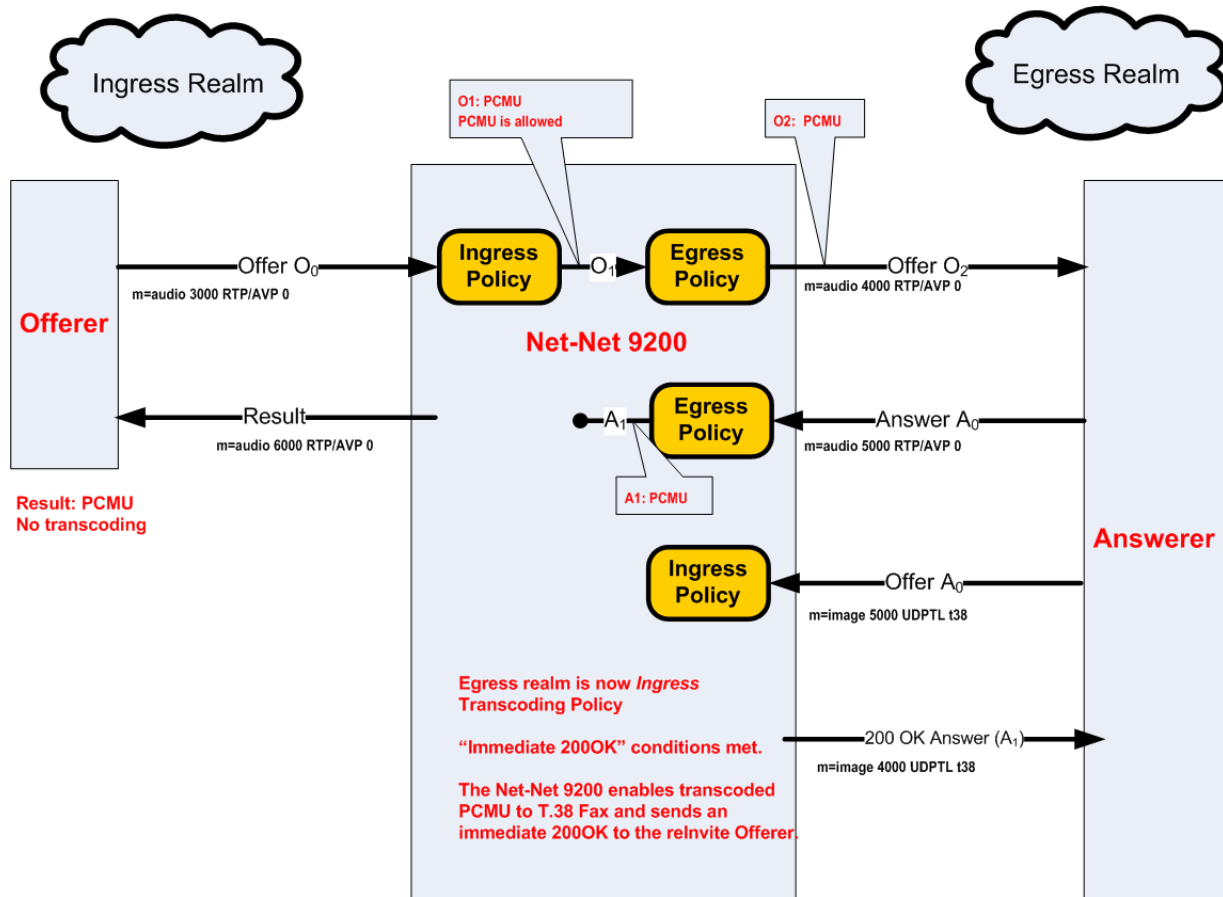
The following scenario represents a special case where a 200OK message is sent directly to the offerer, without passing SDP to the answerer's side if the following conditions are met:

- The SDP represents a Re-Negotiation of Media (i.e., a Re-Invite)
- The SDP contains only one m= line and it is m= image T.38
- The ingress policy allows T.38
- The egress policy disallows T.38 and does not add G711FB
- The answerer side is FAXable

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow	* T.38:no	allow	*
add-on-egress		add-on-egress	
order		order	
force-ptime		force-ptime	
packetization-time		packetization-time	

The following diagram illustrates the Immediate 200 OK for FAX scenario.



Transrating

The Net-Net 9200 can transrate media as it exits the Net-Net 9200 into the network. Transrating is also known as forced packetization time (ptime), and is used to enforce a configured ptime within a realm. Transrating is often desirable when devices in a realm can only accept media with a specific ptime, or to optimize bandwidth.

If this feature is configured, the media portion of a call is transrated regardless of which codecs are ultimately chosen for each realm as long as they are transcodable. This allows realms that have devices that can only use a single packetization interval to interwork with devices that may or may not have the same packetization capabilities.

You must enable force-ptime in the egress transcoding policy and then specify the ptime to force.

Transrating occurs when forced-ptime is enabled and the offered and answered ptimes do not match and the top non-Signaling codec of A_1 and top non-Signaling codec of O_1 are Transcodable.

Note: Answered ptime A_1 does not have to be equal to the ptime inserted into the outgoing offer O_2 , it just has to be different than the offer the Net-Net 9200 received (O_1).

Ptime is configured as an integer between 5 and 240 and must be divisible by 5. Not all codecs support the full range of allowable ptimes, Please refer to [Supported Codecs \(784\)](#) for current list of ptimes per codec supported by the DSPs.

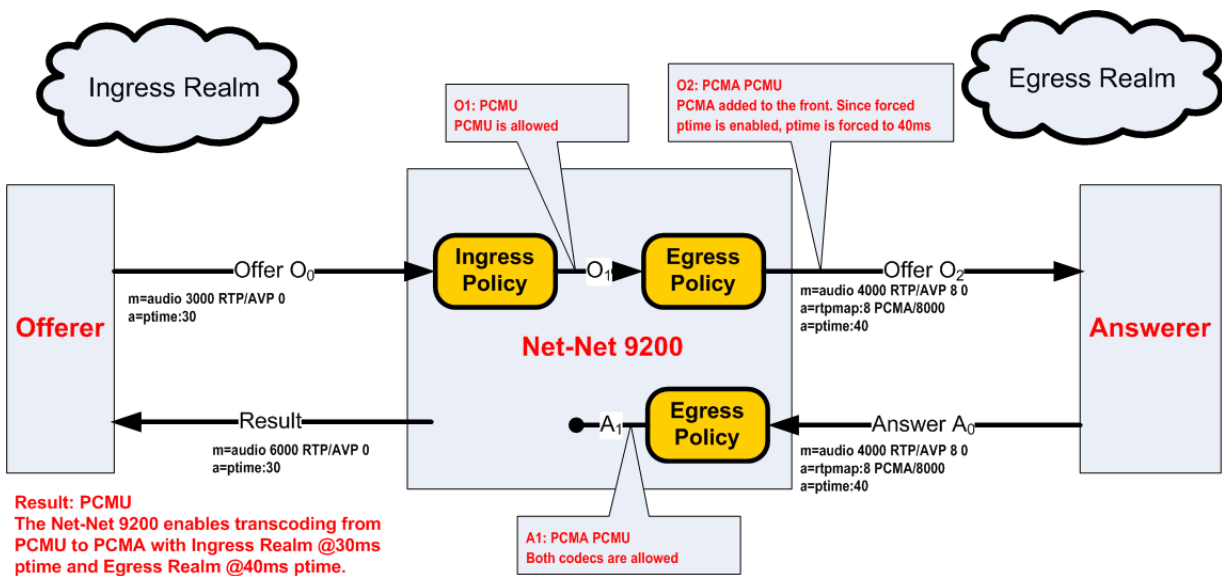
Transrating Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow	*	allow	*
add-on-egress		add-on-egress	PCMA
order	G723 *	order	
force-ptime	disabled	force-ptime	enabled
packetization-time		packetization-time	40

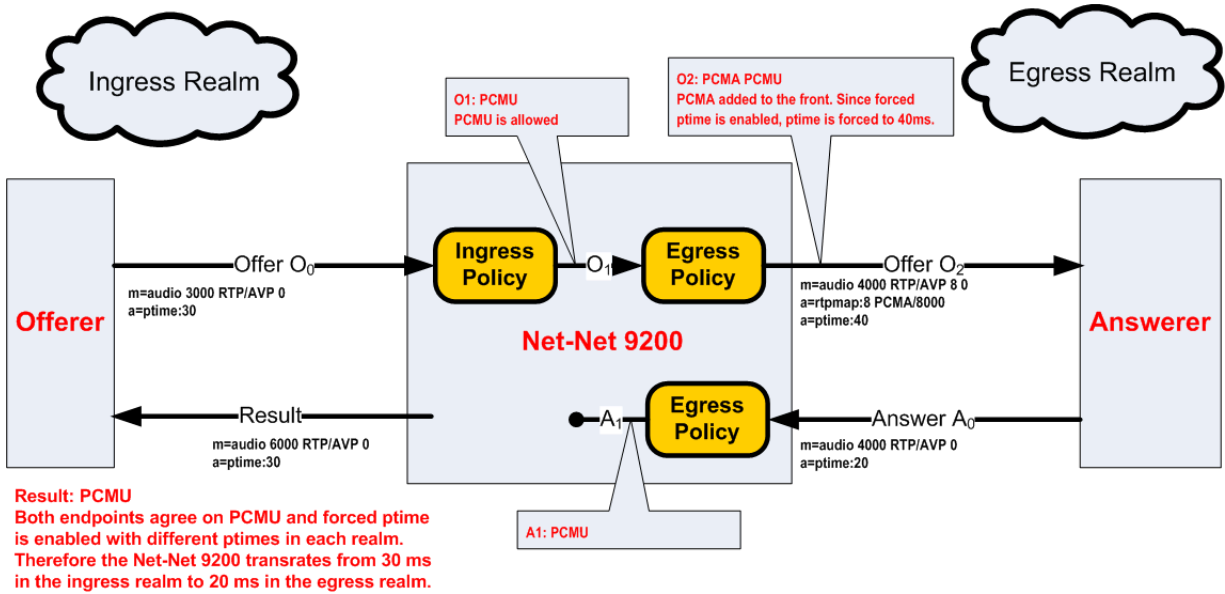
1. In the following diagram, PCMU is offered in the ingress realm with 30ms ptime, and the egress realm is forced to use 40ms ptime. PCMA is added as the top codec for the egress realm.

The Net-Net 9200 enables transcoding between the ingress realm (PCMU) and the egress realm (PCMA) and the ptimes as negotiated are also maintained.



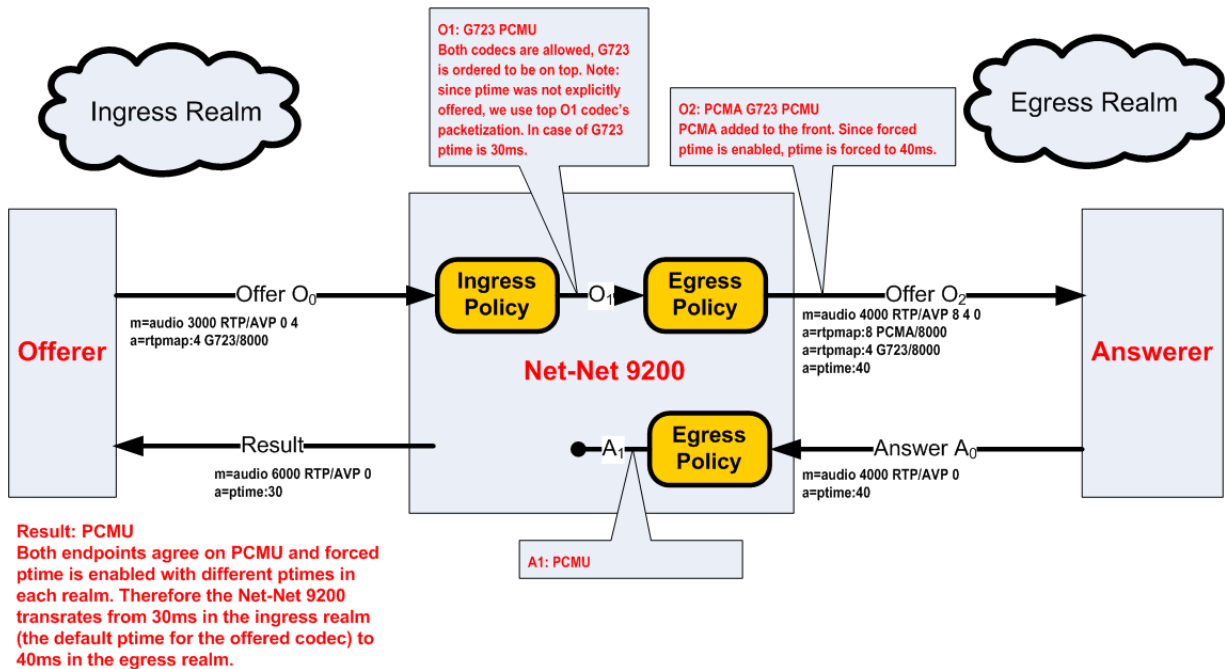
2. In the following diagram, PCMU is offered in the ingress realm with a ptime of 30ms, and forced to 40 ms in the egress realm by policy.

The answerer chooses to use PCMU with a 20 ms ptime. Thus the call is not transcoded, but it is transrated from 30ms in the ingress realm to 20ms in the egress realm.



- In the following diagram, PCMU and G723 are offered in Realm A. The top codec's ptime (30ms) is implied as the one for the ingress realm. The Net-Net 9200 adds PCMA to the SDP offer with a 40ms ptime.

The answerer chooses to use PCMU with a 40 ms ptime. Thus the call is transrated from 30ms in the ingress realm to 40ms in the egress realm.



Default Media Profiles

The Net-Net 9200 contains a set of default media profiles that define characteristics of well-known IANA codecs. You can not view the default media profiles' configurations, but you can override them by configuring identically-named media profile configuration elements.

Transcodable codecs are a subset of the default media profiles which the Net-Net 9200 can transcode between.

Supported Codecs

The following list shows the current default codecs, RTP payload type, media type, and default ptime if available.

Codec	RTP Payload Type	Media Type	Default Ptime
PCMU	0	audio	20
G722FB	0	audio	0
G726-32	2 or dynamic	audio	20
G726	2	audio	20
GSM	3	audio	20
G723	3	audio	30
DVI4	5	audio	
DVI4	6	audio	
LPC	7	audio	
PCMA	8	audio	20
G722	9	audio	20
L16	10	audio	
L16	11	audio	
QCELP	12	audio	
CN	13	audio	
MPA	14	audio	
G728	15	audio	20
DVI4	16	audio	
DVI4	17	audio	
G729	18	audio	20
G729A	18	audio	20
CelB	25	video	
JPEG	26	video	
nv	28	video	
H261	31	video	
MPV	32	video	
MP2T	33	video	
H263	34	video	
G726-16	dyanmic	audio	20
G726-24	dyanmic	audio	20
G726-40	dyanmic	audio	20
iLBC	dyanmic	audio	30
telephone-event	dyanmic	audio	
T.38	t38	image	10

Transcodable Codecs

The following list shows the transcodable codecs which the Net-Net 9200 can add to SDP. These codecs all reflect default media profiles for their given names.

- PCMU
- PCMA
- G729
- G729A
- G729B
- iLBC
- telephone-event
- T.38
- G726
- G726-16
- G726-24
- G726-32
- G726-40
- G723
- GSM FR
- AMR
- EVRC0
- EVRC
- EVRCB

When creating an override media profile from the previously listed codec, case is ignored. Also, GSM FR is entered as "GSM".

Transcodable Codec Details

The following table lists the supported codecs, RTP payload number, default ptime, and supported ptimes.

Codec (supported bit rates)	RTP Payload Type	Default Ptime (ms)	Supported Ptime (ms)
G711 PCMU (64 kbps)	0	20	5, 10, 20, 30, 40, 50, 60
G711 PCMA (64 kbps)	8	20	5, 10, 20, 30, 40, 50, 60
G723.1 (5.3/6.4 kbps)	4	30	30, 60, 90
G729/A/B (8 kbps)	18	20	10, 20, 30, 40, 50, 60, 70, 80
G.726-32 (32 kbps)	2, 96-127	20	5,10,20,30,40,50,60
GSM FR (13 kbps)	3	20	20, 40, 60
G726-16 (16 kbps)	96-127	20	5,10,20,30,40,50,60
G726-24 (24 kbps)	96-127	20	5,10,20,30,40,50,60
G726-40 (40 kbps)	96-127	20	5,10,20,30,40,50,60
iLBC-15.20K (15.2 kbps)	96-127	20	20,240
iLBC-13.33K (13.33 kbps)	96-127	30	30,240

H.323 Transcodable Codecs

An H.323 call can be transcoded if it uses one of the following codecs: G711, G.723, G726, G.728, G.729.

Preferred Default Payload Type

When the Net-Net 9200 adds a codec with a dynamic payload type to SDP, it uses the lowest unused payload number. You can configure a preferred payload type for a dynamic codec by creating an override media profile. This makes the Net-Net 9200 use your preferred payload type for insertion into SDP. If you configure a dynamic codec to use a preferred payload type, and that payload type is already in use, the codec will still be inserted into SDP, but with the first available dynamic payload type.

For example, you create a media profile for telephone-event with a payload type of 101. If telephone-event is added to SDP, and payload type 101 is already in use in the SDP, the Net-Net 9200 will use the first available payload type in the 96-127 range when adding telephone-event.

Redefining Codec Packetization Time

You can configure a media profile with a packetization time (ptime) that overrides the codec's default ptime. Transcoding functions look up and use default ptimes when not specified in offered or answered SDP. Default ptime for most audio codecs is 20ms; some however are 30ms. See the [Transcodable Codec Details \(785\)](#) list for default values.

To change the default ptime for a codec, you must create a media profile that overwrites the default ptime parameter with your new packetization time. When SDP is received with no 'a=ptime' attribute or when adding the codec to egress SDP, the newly configured ptime is used.

New default ptime for a media profile is entered by typing "ptime=<x>" in the parameters parameter, where <x> is the new default packetization time.

Note: The frames per packet parameter in the media profile configuration element has no affect on a codec's ptime. This parameter is used for H.323 functionality.

mptime Support for Packet Cable

The SDP specification lacks the ability to specify unique packetization times per codec when more than one codec is listed in an m= line. The ptime attribute is not related to a specific codec but to the entire m= line. When multiple codecs appear on a single m= line, the PacketCable mptime attribute can specify different packetization times for each codec.

The Net-Net 9200 adheres to PKT-SP-NCS1.5-I01-050128 and PKT-SP-EC-MGCP-I06-021127 for processing and generating mptime. The mptime line uses an integer to indicate the packetization time for each corresponding codec in the m= line. The dash character, "-", on an mptime line is used for non-packetized codecs, such as CN or telephone-event.

If the Net-Net 9200 receives an invalid mptime, it is ignored and removed. If a valid mptime is received in the incoming SDP, its values will be used for packetization times of each corresponding codec and a valid mptime line will be sent in the outgoing SDP.

Valid:

```
m=audio 10000 RTP/AVP 0 96 8
a=mptime:20 - 30
a=rtpmap:96 telephone-event/8000
```

Valid: 'ptime' attribute is ignored

```
m=audio 10000 RTP/AVP 0 8
a=mptime:20 30
a=ptime:30
```

Invalid: dash cannot be first mptime value

```
m=audio 10000 RTP/AVP 96 0
a=mptime: - 20
```

When Net-Net SBC includes an mptime in an outgoing SDP, it will also always add a ptime attribute with the value of the most preferred codec. This is done to increase the interoperability with devices that do not support mptime.

Redefining G711FB

G711 Fall Back (G711FB) is an internal codec that encompasses all FAXable codecs. The two current FAXable codecs are PCMU and PCMA. G711FB defaults to PCMU when transcoding policy needs to add a FAXable codec in SDP. You can change G711FB to offer PCMA by configuring the media-profile for appropriately. G711FB is only used for FAX transcoding scenarios.

To redefine G711 FB, create a media profile configuration element named g711fb and set the payload-type to 8.

AMR-NB Specifications

The Net-Net 9200 supports Adaptive Multi-Rate Narrow Band (AMR-NB) codec. All configurations of this codec, as indicated by SDP, are transcodable except when the following SDP parameters are enabled:

- robust-sorting
- interleaving

When AMR is configured in a transcoding policy's add-on-egress parameter, it is forwarded from the Net-Net 9200 with the following default settings:

- 12.2 kbps
- RTP/IF1 format
- No redundant packets
- Bandwidth efficient payload
- No CRC frame
- 20ms default ptime

Note: This codec requires a license.

EVRC Codec Family

The Net-Net 9200 supports 2 codecs in the EVRC family. Each codec has has three variants.

- EVRC-A is also commonly referred to as EVRC. The following EVRC-A packet formats are supported:

Header-freepacket format = EVRC0

Bundled/interleaved packet format = EVRC

Compact Bundled packet format = EVRC1

- The following EVRC-B packet formats are supported:

Header-free packet format = EVRCB0

Bundled/interleaved packet format = EVRCB

Compact Bundled packet format = EVRCB1

Note: Both EVRC codecs require their own license.

EVRC-A Specifications

The Net-Net 9200 supports the Enhanced Variable Rate codec (EVRC) for transcoding. Three subtypes of the EVRC codec are supported as media-profiles:

1. EVRC0—header-free EVRC format. This codec is transcodable.
2. EVRC—bundled/interleaved packet format. This codec is transcodable.
3. EVRC1—fixed-rate compact bundled packet format. This codec is transcodable.

To transcode to and from EVRC, the EVRC license must be installed.

Note: The Net-Net 9200 handles optional parameters according to RFC 4788 unless otherwise specified.

EVRC0 Supported Options

Required Parameters: none

Optional Parameters:

- **silencesupp**—If this parameter is not present in a DTX session, the default value 1 MUST be assumed.
- **dtxmax**—If this parameter is not present in a DTX session, the default value is 32.
- **dtxmin**—If this parameter is not present in a DTX session, the default value is 12.
- **hangover**—If this parameter is not present in a DTX session, the default value 1 MUST be assumed.

The payload type is dynamic for this codec. EVRC0 specifies the header-free format of the EVRC codec. Only a single frame (20 ms) is allowed in header-free mode.

EVRC Supported Options

Required Parameters: none

Optional parameters:

- **ptime**
- **maxptime**
- **maxinterleave**—If not signaled, the maxinterleave length is set to 5.
- **silencesupp**—If this parameter is not present, the default value 1 MUST be assumed.
- **dtxmax**—See dtxmax in EVRC0 section.
- **dtxmin**—See dtxmin in EVRC0 section.
- **hangover**—See hangover in EVRC0 section.

The payload type is dynamic for this codec. The default ptime is 20 ms. The ptimes 20, 40, and 60 ms are supported for transcoding.

EVRC1 Supported Options

Required parameters: none

Optional parameters:

- **ptime**—See RFC 4566
- **maxptime**
- **fixedrate**—Valid values are 0.5 or 1. If this parameter is not present, 0.5 is assumed.
- **silencesupp**—If this parameter is not present, 1 MUST be assumed.
- **dtxmax**—See dtxmax in EVRC0 section.
- **dtxmin**—See dtxmin in EVRC0 section.
- **hangover**—See hangover in EVRC0 section.

The payload type is dynamic for this codec. Only the 20 ms ptime is supported for transcoding. The media rate will be fixed to either full or half rate depending on the fixedrate parameter (half rate is default).

Default settings for EVRC-A encoding:

- CDMA Rate change for Dim and Burst disabled
- CDMA DTX control enabled

EVRC-A Configuration

In the ACLI, refer to EVRC codecs as follows:

```
EVRC0
EVRC
EVRC1
```

In the ACLI, refer to EVRC codecs' parameters exactly as they appear in the above list.

EVRC-B Specifications

The Net-Net 9200 supports the Enhanced Variable Rate codec, extension B (EVRCB) for transcoding. Three subtypes of the EVRCB codec are supported as media-profiles:

1. EVRCB0—header-free EVRC format. This codec is transcodable.
2. EVRCB—bundled/interleaved packet format. This codec is transcodable.
3. EVRCB1—fixed-rate compact bundled packet format. This codec is transcodable.

To transcode to and from EVRCB, the EVRCB license must be installed.

Note: The Net-Net 9200 handles optional parameters according to RFC 4788 unless otherwise specified.

Note: Installing the EVRC-B license reduces transcoding capacity between some codec pairs. See [Transcoding Capacity \(758\)](#).

EVRCB0 Supported Options

EVRCB0 is the header-free format of the EVRCB codec. Only a single frame (20 ms) is allowed in header-free mode.

Required Parameters: none

Optional Parameters:

- **silencesupp**—If this parameter is not present in a DTX session, the default value 1 MUST be assumed.
- **dtxmax**—If this parameter is not present in a DTX session, the default value is 32.
- **dtxmin**—If this parameter is not present in a DTX session, the default value is 12.
- **hangover**—If this parameter is not present in a DTX session, the default value 1 MUST be assumed.

The payload type is dynamic for this codec. EVRCB0 specifies the header-free format of the EVRCB codec. Only a single frame (20 ms) is allowed in header-free mode.

EVRCB Supported Options

Required Parameters: none

Optional parameters:

- **ptime**
- **maxptime**
- **maxinterleave**—If not signaled, the maxinterleave length is set to 5.
- **silencesupp**—If this parameter is not present, the default value 1 MUST be assumed.
- **dtxmax**
- **dtxmin**
- **hangover**

The payload type is dynamic for this codec. The default ptime is 20 ms. Only the 20 ms ptime is supported for transcoding.

EVRCB1 Supported Options

Required parameters: none

Optional parameters:

- **ptime**—See RFC 4566
- **maxptime**
- **fixedrate**—Valid values are 0.5 or 1. If this parameter is not present, 0.5 is assumed.
- **silencesupp**—If this parameter is not present, 1 MUST be assumed.
- **dtxmax**
- **dtxmin**
- **hangover**

The payload type is dynamic for this codec. Only the 20 ms ptime is supported for transcoding. The media rate will be fixed to either full or half rate depending on the fixedrate parameter (half rate is default).

Default/fixed settings for EVRC-B encoding:

- CDMA Rate change for Dim and Burst disabled
- CDMA Timing alignment disabled
- CDMA DTX control enabled
- EVRC-X Delay reduction disabled

EVRC-B Configuration

In the ACLI, refer to EVRCB codecs as follows:

EVRCB0
EVRCB

EVRCB1

In the ACLI, refer to EVRC codecs' parameters exactly as they appear in the above sections in the bulleted lists.

Configuring Transcoding

TCU Configuration

Based on the number of TCUs available, you can configure your transcoding hardware for redundancy or more capacity. The `tcu-double-active` parameter sets the boot state of the TCU in slot 6 as enabled or disabled. Redundancy configuration follows from there according to the table below. Changing the `tcu-double-active` state requires a system reboot to take effect. Refer to the following table:

TCUs	<code>tcu-double-active</code> configuration	Initial TCU HA State
1	enabled or disabled	no effect; 1 active TCU
2	enabled	2 + 0
2	disabled	1 + 1
3	enabled	2 + 1
3	disabled *	1 + 1

Caution

* Acme Packet recommends against disabling TCU-double-active with three TCUs installed in the Net-Net 9200. To run your Net-Net 9200 in 1+1 TCU HA configuration, with 3 TCUs on-hand, remove the TCU in slot 6, and replace it with a blank bezel.

ACLI Instructions and Examples

To set transcoding hardware configuration:

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


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


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


```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```
4. `tcu-double-active`—Set this parameter to enabled for TCUs in slots 4 and 5, when populated, to both run in active mode.
5. Save and activate your configuration. You must reboot the Net-Net 9200 for this parameter to take effect.

Transcoding Policy Configuration

Transcoding is configured by creating transcoding policies and referencing them from a realm configuration.

ACLI Configuration Instructions and Examples

The parameters that you can configure are name, allow, add-on-egress, order, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Note: A single transcoding policy can be reused for any number of realms.

To access the configuration parameters for transcoding policies:

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

```
ACMEPACKET# configure terminal
```

2. Type `media-manager` and press <Enter>.

```
configure# media-manager
```

3. Type `transcoding-policy` and press <Enter>.

```
media-manager# transcoding-policy
```

From this point, you can start configuring your transcoding policy.

Naming Transcoding Policies

The transcoding policy's name is important not only because it uniquely identifies the policy, but because it is the name you will enter into your realm configuration's **transcoding-policy** parameter. It is important to apply the correct policy to the appropriate realm.

To set the transcoding policy's name:

1. **name**—Set the name for this transcoding policy, and note it for future reference when you apply transcoding policies to realms. This parameter is required, and has no default.

```
transcoding-policy# name private
```

Removing, Allowing, and Adding Codecs

The Net-Net 9200 removes and allows codecs using the **allow** parameter. Refer to the [Transcoding Policy Definition \(762\)](#) section of this chapter for configuration information.

- **Allow**—The **allow** parameter takes a list of codecs that you want to pass through the Net-Net 9200 and can explicitly allow them to remain in the SDP for the next step; codecs not matching the items on this list are removed. This parameter is required.
- **Add-on-egress**—The **add-on-egress** parameter sets the codecs that the Net-Net 9200 adds to an offer if that codec is not already there. This parameter applies only to the egress policy.

For allow, order, and add codecs to transcoding policies:

You can configure and edit these two transcoding parameters as ACLI lists, meaning that there are **add** and **delete** commands associated with each. You type the name of the parameter, choose the operation you want to perform on the list (adding or deleting), and then specify the data that you want to add or remove.

The examples in the procedure that follows show you how to add to the lists you are configuring. To remove items from the **allow** list, simply replace the **add** command you see in these example with **delete** and the items you want to remove.

If you want to overwrite previous values, you can enter the command, a <Space> and the items in the list.

1. **allow**—Enter a list of codecs that are allowed to pass through the Net-Net 9200. To allow all codecs, enter an asterisk (*).

```
transcoding-policy# allow *
```

2. **add-on-egress**—Enter the codecs that you want added to the SDP offer for the egress transcoding policy. If you leave this parameter blank, then the Net-Net 9200 will not add codecs to the SDP answer. This parameter cannot be wildcarded; other possible value are listed in the [Supported Codecs \(784\)](#) section of this chapter.

```
transcoding-policy# add-on-egress add G729
```

Ordering Codecs

Transcoding policy can specify the order that codecs appear in the SDP offer or answer. Refer to the [Transcoding Policy Definition \(762\)](#) section of this chapter for configuration information.

To configure an order which codecs appear in the offer:

1. **order**—Enter the order in which you want codecs to appear in the SDP offer or answer. You can enter them in any of the ways described in the preceding explanation.

```
transcoding-policy# order G711 * G729
```

Transrating Configuration

The following procedure explains how to configure transrating for a transcoding policy. This transcoding policy must be applied as an egress transcoding policy.

To configure forced ptime for a transcoding policy:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **transcoding-policy** and press <Enter>.


```
ACMEPACKET(media-manager)# transcoding-policy
ACMEPACKET(transcoding-policy)#
```
4. If you are adding support for this feature to a pre-existing configuration, then you must select the specific configuration instance, using the ACLI **select** command.


```
ACMEPACKET(transcoding-policy)# select 1
```

You can now configure forced ptime.
5. **force-ptime**—Set this parameter to **enabled** to enable forced ptime for this transcoding policy.
6. **packetization-time**—Enter the ptime in ms to use in the realm where this transcoding policy is active.
7. Save your work using the ACLI **done** command.

Applying a Transcoding Policy to a Realm

Once you have configured a transcoding policy, you apply it to a realm by configuration name.

To apply a transcoding policy to a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type `media-manager` and press <Enter>.


```
configure# media-manager
```
3. Type `realm-config` and press <Enter>.


```
media-manager# realm-config
```
4. **transcoding-policy**—Enter the name of the transcoding policy that you want to apply to this realm. This value is the same as the one you entered in the name parameter for the transcoding policy you want to use for this realm. There is no default for this parameter.


```
realm-config# transcoding-policy private
```

Transcoding in the Same Realm

If you want the Net-Net 9200 to transcode media between endpoints within a single realm, you must enable the `transcode in realm` and the `media manage in realm` parameters.

To transcode within a realm:

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


```
ACMEPACKET# configure terminal
```
2. Type `media-manager` and press <Enter>.


```
configure# media-manager
```
3. Type `realm-config` and press <Enter>.


```
media-manager# realm-config
```
4. **transcode-in-realm**—This parameter enables or disables transcoding policies for the realm. The default is `disabled`.


```
realm-config# transcode-in-realm enabled
```
5. **mm-in-realm**—Ensure this parameter is set to `enabled` to use transcoding within the realm. The default is `disabled`.


```
realm-config# mm-in-realm enabled
```
6. Save your work using the ACLI **done** command.

Transcoding in the Same Network

If you want transcoding media between endpoints within a network, you must enable the `transcode in network` and the `media manage in network` parameters.

To transcode within a network:

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


```
ACMEPACKET# configure terminal
```
2. Type `media-manager` and press <Enter>.


```
configure# media-manager
```

3. Type **realm-config** and press <Enter>.


```
media-manager# realm-config
```
4. **transcode-in-network**—This parameter enables or disables transcoding within a single network. The default is **enabled**.


```
realm-config# transcode-in-network disabled
```
5. **mm-in-network**—Ensure this parameter is set to **enabled** to use transcoding within the network. The default is **enabled**.


```
realm-config# mm-in-network enabled
```
6. Save your work using the ACLI **done** command.

Media Profile Configuration

Media profiles must be created and then defined when you want to override the Net-Net 9200's default media profiles.

ACLI Configuration Instructions and Examples

The parameters that you can configure are name, allow, add-on-egress, order, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Creating User-Defined Ptime per Codec

To change the Net-Net 9200's default ptime for a specific codec, you must create a media profile configuration element. In the **parameter** parameter, you set the ptime to the value of your choosing.

Note: The frames-per-packet parameter in the media profile configuration element is NOT used for setting a user defined ptime for that codec.

To configure a new ptime value for a codec:

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


```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter>.


```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **media-profile** and press <Enter>.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

If you are adding ptime to a pre-existing media profile, then you must select (using the ACLI select command) the configuration that you want to edit. If you are adding ptime to an undefined media profile, you must create it first.
4. **name**—Type the name of the codec for which you are creating a new default ptime.


```
ACMEPACKET(media-profile)# name pcmu
```
5. **payload-type**—Enter the well-known payload type for this codec.


```
ACMEPACKET(media-profile)# payload-type 0
```
6. **parameter**—Set the ptime by typing **parameter**, a <Space>, **ptime=**, the new ptime value. Then press <Enter>. For example:


```
ACMEPACKET(media-profile)# parameter ptime=40
```
7. Save your work using the ACLI **done** command.

ReINVITE for Fax Calls

In some deployments, an originator sends inband fax messages through the Net-Net SBC to terminating endpoints that do not support uncompressed codecs. Thus the terminating call leg must communicate FAXes either through out of band T.38 or in-band G.711 codecs. In some cases the terminating endpoint can determine that it is being sent a FAX and send a reinvite to request that it be sent T.38 FAX instead of inband FAX, thereby switching from an audio call to a FAX call. If the Net-Net SBC does not receive this reINVITE, it will send its own reinvite toward the terminating endpoint to establish the FAX session with a codec the endpoint can support.

FAX Call Detection

The Net-Net SBC detects fax tones based on the receipt of the DIS Preamble V.21 flag which is sent after the CNG/CED tones are exchanged. These tones are embedded in a faxable codec in the RTP media stream. You must set the tone-detection parameter in the transcoding policy to FAX as the first step in configuring this feature.

Note: Enabling tone-detection steers all faxable calls through the DSPs thus requiring DSP resources.

Renegotiate Timer

Upon detection of the DIS Preamble V.21 flag, the Net-Net SBC starts a configurable tone Detect Renegotiate Timer. The **tone detect renegotiate timer** parameter is configurable per transcoding policy and defaults to 500ms. If the Net-Net SBC receives a ReINVITE from the originating or terminating endpoint, the tone Detect Renegotiate Timer will be reset and no ReINVITE will be sent from the Net-Net SBC toward the terminating endpoint.

If the Net-Net 9200 does not receive a ReINVITE from the called endpoint before this timer expires, it creates and sends an INVITE to the terminating endpoint.

ReINVITE Codecs

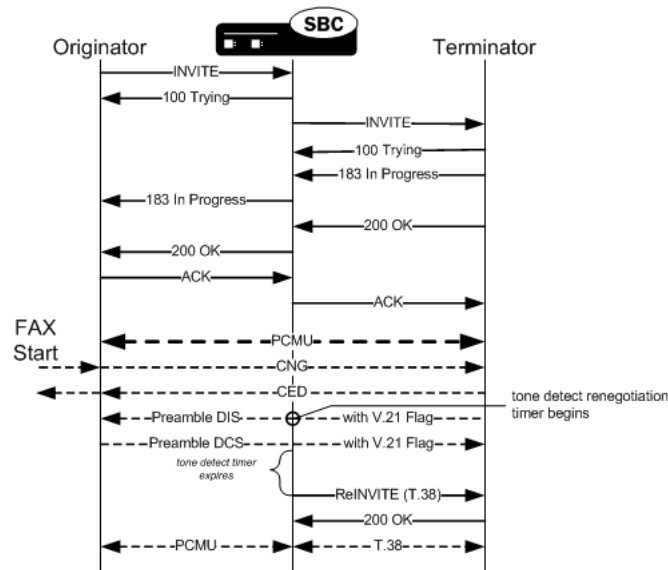
There are two codecs that can be inserted into the reINVITE message: T.38 & G711 (defaulting to PCMU). You configure **T.38OFD** and/or **G711OFD** in the add codecs on egress parameter in the transcoding policy configuration element. These codec names are only used for reINVITEs in this feature. Each one is inserted into the reINVITE's message body on its own m= line.

In response to the terminator that sends a 488 Response to the Net-Net SBC's ReInvite bearing aT.38OFD SDP offer, the Net-Net SBC can send a second ReINVITE called and configured as **OFDFB**. OFDFB, On Fax Detect Fallback is G711 with a default PCMU codec that can be overridden in a media profile to request PCMA. The OFDFB codec will never be added in an initial offer.

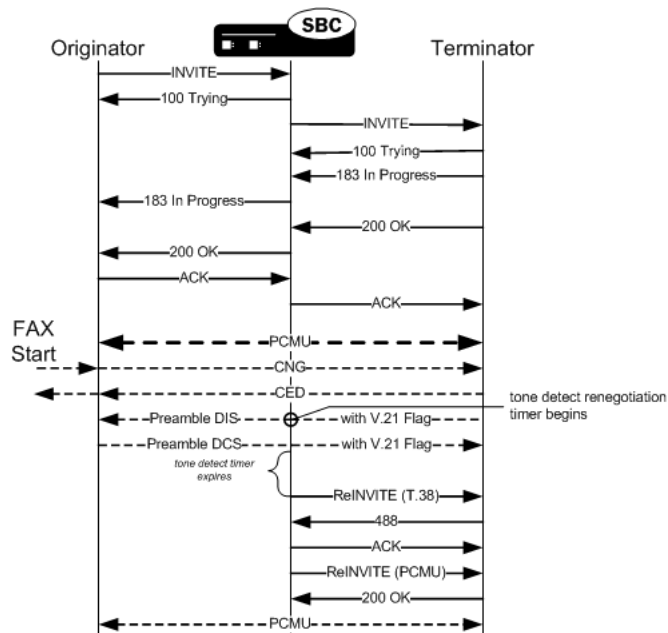
Call Completion

In the typical case, the Net-Net SBC sends a reinvite toward the terminating endpoint with the specified codecs. The terminating endpoint replies with a 200OK indicating it can use the specified codec. When this call leg is set up, FAX media will flow through the Net-Net SBC and be transcoded between the originator's codec and the terminator's codec.

The following call flow shows the typical example:



When the terminating endpoint rejects the T.38 reINVITE, and the OFDFB is configured as an add-on-egress codec, the Net-Net 9200 sends a G711 offer toward the call terminator.



Glare Condition

After not receiving a ReINVITE from either of the endpoints after the configured amount of time, the Net-Net SBC sends a ReINVITE toward the terminating endpoint. If the Net-Net SBC receives a ReINVITE from either endpoints while waiting for the response to its own ReINVITE, it will reject the endpoint-sourced ReINVITE with a 491 Request Pending error message.

The glare condition produces an internal error message, 877, that by default is mapped to a 491 message.

ACLI Instructions

To configure a transcoding policy for reINVITES for FAX calls:

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


```
ACMEPACKET# configure terminal
```
2. Type media-manager and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type transcoding-policy and press <Enter>.


```
ACMEPACKET(media-manager)# transcoding-policy
```
4. Use the ACLI **select** command to select a transcoding policy.


```
ACMEPACKET(transcoding-policy)# select
```

You may now configure a transcoding policy that will create reINVITES for FAX calls.
5. **add-codecs-on-egress**—Set this parameter to either **T.38OFD** and/or **G711OFD** depending on the value the m= line should have on reINVITE. You can additionally add the **OFDFB** value to this parameter to enable the second fallback case.
6. **tone-detect-renegotiate-timer**—Set this parameter to a value between 50 and 3200 ms.
7. **tone-detection**—Set this parameter to FAX to use the ReINVITE for FAX calls feature.
8. Save and activate your configuration.

G.726 transcoding with H.323

The Net-Net 9200 supports transcoding between G726 and G711 (PCMU and PCMA), G723, G726, G728, and G729 for H.323 to H.323 calls. In this scenario, the system supports G726 at four transmission speeds.

Max Bit rate	Data Rate	Codec specified in configuration
160	16 kbit/s	G726-16
240	24 kbit/s	G726-24
320	32 kbit/s	G726-32
400	40 kbit/s	G726-40

Media Profile Configuration

For any H.323 transcoding scenarios you must first create a media profile named genericAudioCapability that accepts all default values. For example:

```
media-profile
  name genericAudioCapability
  subname
  media-type audio
  payload-type
  transport RTP/AVP
  req-bandwidth 0
  frames-per-packet 0
  parameters
  average-rate-limit 0
```

```

max-burst-size          0
peak-rate-limit         0
sdp-rate-limit-headroom 0
sdp-bandwidth           disabled

```

G726-16, G726-24, or G726-40 use dynamic media type values and no configuration is necessary unless you wish to assign a specific media type to a given codec.

Transcoding Policy Configuration

The process and execution of H323 transcoding policies is similar to that of SIP. For G726 transcoding, there are two guidelines:

- To configure the add-on-egress parameter with a G726 code, values, the value must be specified as G726-xx.
- To configure the allow-codecs parameter with a G726 codec, the value must be specified as G726-xx.

For example:

```

transcoding-policy
  name          all-all-add-G726
  allow-codecs   *
  add-codecs-on-egress  G726-40
  order-codecs
  force-ptime    disabled
  packetization-time 20
  dtmf-in-audio  disabled
transcoding-policy
  name          only-G726
  allow-codecs   G726-40
  add-codecs-on-egress  G726-40
  order-codecs
  force-ptime    disabled
  packetization-time 20
  dtmf-in-audio  disabled

```

G726 OLC Format

When a G726 codec is configured in the **add on egress** parameter, the following OLC message is sent from the Net-Net 9200 toward the H.323 endpoint. This example shows when the Net-Net 9200 adds G726-40, and is identifiable from the value of the maxBitRate parameter:

```

OpenLogicalChannel
  forwardLogicalChannelNumber: 3
  forwardLogicalChannelParameters
    dataType: audioData (3)
      audioData: genericAudioCapability (20)
        genericAudioCapability
          capabilityIdentifier: standard (0)
            standard: 0.0.7.726.1.0 (itu-t recommendation
7 726 1 0)
            maxBitRate: 400
      multiplexParameters: h2250LogicalChannelParameters (3)
  reverseLogicalChannelParameters
    dataType: audioData (3)
      audioData: genericAudioCapability (20)
        genericAudioCapability

```

```

                                capabilityIdentifier: standard (0)
                                standard: 0.0.7.726.1.0 (itu-t recommendation
7 726 1 0)
                                maxBitRate: 400
multiplexParameters: h2250LogicalChannelParameters (2)

```

Media Type Subnames

You can define multiple versions of a media profile for a single codec by using the subnames feature. You can then reference the new media profile by a combination of the media profile name and media profile subname.

Some media types are not unique per just their value in an SDP **m=** line, they must be uniquely identified by looking at additional SDP parameters. For example, you can define a media profile for G729, when only the parameter and value **annexb=yes** is present in the SDP. By creating a media profile + subname that defines both a media type and parameter, you can perform various operations on G729 only when **annexb=yes** is encountered.

Some applications of media type subnames are:

- maintaining different versions of the same codec with different bandwidth ceilings
- maintaining different versions of the same codec with different ptimes
- grouping codecs by using customer as a subname
- grouping codecs by using realm as a subname

SDP Parameter Matching

This feature matches parameters in the **a=fmtp**, codec-specific SDP **a=** line. It does not try to match a global **m=** line attribute like **a=p-time**.

Using Subnames with Transcoding Policies

Media profiles are defined and referenced in the ACLI by a name and subname in the following format

```
<name>::<subname>
```

If no subname has been created for a media profile, you may continue using the media profile name without any subname specifier.

For example, to remove a media profile and subname configured as **PCMU::customer1** from all SDP entering the egress realm, you would configure the transcoding policy **allow** parameter as follows:

```
allow PCMU::customer1:no
```

Subname Syntax and Wildcarding

You can wildcard one or both portions (name and subname) of a media type and subname pair:

- When you wildcard the **name** portion of the value, you can provide a specific subname that the Net-Net SBC uses to find matching media profiles.
- When you wildcard the subname portion of the value, you can provide a specific **name** that the Net-Net SBC uses to find matching media profiles.

The following table defines and explains subname wildcarding and syntax:

Syntax	Example Value	Description
<name>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <name>::*.
<name>::	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<name>::*	PCMU::*	Matches any and all media profiles with the name value configured as PCMU with any and all subname configured.
<name>::<subname>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
*	*	Matches anything, but does not have to be a defined media profile.
::	*::*	Matches any and all media profiles, but requires the presence of media profile configurations.
*::<subname>	*::64k	Matches all media profiles with this subname. You might have a group of media profiles with different names, but the same subname value.
*::	*::	Matches any media profiles with an empty subname parameter.
::	::	Invalid
::*	::*	Invalid

Wildcarding add-on-egress

It is important to note that you may not configure **add-on-egress** with a wildcarded subname in a transcoding policy. You may only add a specific instance of a media type.

Valid:

```
add-on-egress PCMU
add-on-egress PCMU::customer1
```

Invalid:

```
add-on-egress PCMU::*
```

ACLI Instructions and Examples

To use media type subnames with a transcoding policy, you must first configure a media profile and subname. Then you can configure a transcoding policy with a media type and subname pair for your application

To use configure a media type and subname:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **media-profile** and press <Enter>.


```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **name**—Type the name of the codec for which you are creating a new default ptime.


```
ACMEPACKET(media-profile)# name pcmu
```
5. **subname**—Enter a description for the use of this subname


```
ACMEPACKET(media-profile)# subname annexa=yes
```

You may now configure this subname's unique attributes. PCMU is created with ptime of 30 in this example.
6. **parameter**—Set the ptime by typing **parameter**, a <Space>, **ptime=**, the new ptime value. Then press <Enter>. For example:


```
ACMEPACKET(media-profile)# parameter annexa=yes
```

Note: Remember to configure all additional, required media profile parameters, or they will inherit default values.
7. Save your work using the ACLI **done** command.

To configure a transcoding policy with a media type with subname:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
```
3. Type **transcoding-policy** and press <Enter>.


```
ACMEPACKET(media-manager)# transcoding-policy
```
4. Use the ACLI **select** command to select a transcoding policy.


```
ACMEPACKET(transcoding-policy)# select 1
```

You may now enter a media profile with subname to any parameter in the transcoding policy that accepts a media profile.
5. **allow**—Enter a list of codecs that this transcoding policy allows or denies from passing through the Net-Net 9200. To allow all codecs, enter an asterisk (*).


```
transcoding-policy# allow pcmu::annexa=yes:no
```
6. Save and activate your configuration.

Diameter: CAC/RACF

The Diameter base protocol (RFC 3588) is supported by the Net-Net SBC and is used for Bandwidth-Based Call Admission Control (CAC) applications. You need to install the External BW Mgmt, and External Policy Services licenses to enable this feature. Please contact tac@acmepacket.com to receive your software licences.

FQDN as Address for External Policy Server

You can configure an external policy servers' address as an FQDN in addition to an IP address.

FQDN Configuration

FQDNs are configured in the address parameter in the external policy server configuration element. These FQDNs must conform to RFC 1035. If the port parameter in external policy server configuration element is not zero, then it will be used to connect to the group of applicable external policy servers. If the port parameter is set to zero, then the port number returned in SRV RR from the DNS server will be used. When the Net-Net SBC queries a DNS server for an FQDN configured in the external policy server, it always appends the `_diameter._tcp.` to request only TCP based Diameter servers.

Diameter Policy Server High Availability

External Policy Server HA Cluster

The Net-Net SBC can provide external policy server redundancy through a combination of multiple servers being returned in one FQDN query and maintaining state of these servers.

When multiple IP addresses are returned in a response to a DNS query for Diameter-based external policy servers, the Net-Net SBC assembles the IP addresses into an HA cluster which provides redundancy for Diameter-based applications. This feature is enabled by configuring the external policy server's **address** parameter with an FQDN.

The number of servers maintained in the HA cluster is configured with the **max connections** parameter. Thus if the **max connections** parameter is set to 3, the Net-Net SBC maintains 1 active external policy server with 2 back up servers.

Standby Server Prioritization

The Net-Net SBC looks at the priority and weight fields in the DNS response to create the preferred order of the the primary, secondary, tertiary, and quaternary Policy Servers. These 4 addresses are known as the top-level PSs in the cluster.

The Net-Net SBC uses the priority and weights according to RFC 2782. However, weights only apply when multiple servers share the same priority value. If the priorities are different, then the Net-Net SBC will not use weights. If the priorities are the same, then weight of the two (or more) contested servers is used to determine which one to use.

Diameter Server States

An HA cluster can contain up to 4 Policy Servers, where the TCP/Diameter connection is established and monitored. Diameter session traffic is only sent to the active policy server in the cluster. The policy servers exist in the following states:

- active-TCP and Diameter connection established. Net-net SBC using this server for policy decisions. The policy server with the highest priority/weight begins in this state.
- standby- TCP and Diameter connection established. Server in standby mode.
- inactive - Diameter connection not successfully established. Net-Net SBC tries to reconnect to inactive servers.

HA Cluster Refresh

The Net-Net SBC sends followup SRV queries to the DNS server to refresh the list of available policy servers in the cluster in the following instances:

- DNS cache expires after the TTL is exceeded
- a new policy server with FQDN for an address is configured, saved, and activated on the Net-Net SBC
- after an SPU switchover, the newly active SPU performs a DNS query

When the Net-Net SBC re-queries the DNS server for Diameter external policy servers, the cluster is refreshed with the new/changed servers. Policy server priority is also reconfigured based on newly returned priorities and weights. Upon a cluster refresh, the Net-Net SBC:

- closes connections with standby policy servers that are no longer in the cluster
- creates connections with policy servers which are new in the set

If the currently active policy server remains a member of the cluster after a refresh, it remains active even if its priority has changed. If the active-before-the-refresh policy server is not a member of the cluster after a DNS refresh, the Net-Net SBC gracefully closes the connection to this server. The Net-Net SBC then installs the highest priority server as the active policy server

DNS Failure

If the Net-Net SBC fails to get a response from the DNS server or does not receive at least one IP address in the SRV RR, it continues to send SRV queries periodically, starting with 5 seconds and doubling the interval for every sequential failure, until it receives a valid response. While waiting for a successful DNS response, the Net-Net SBC uses the existing Diameter servers in the DNS cache.

Policy Server Failover

The active external policy server fails over to the highest priority standby server when:

- the TCP connection is closed due to a RST or FIN
- the Diameter connection (CER/CEA exchange) could not be established
- a configured number of consecutive Diameter message timeouts occur. This number is configured in the **max timeouts** parameter. The max timeouts parameter refers to all Diameter messages except for the following:
- You can configure Diameter keepalive message time-outs separately from all other Diameter messages by setting **watchdog ka timer** in the external policy server configuration element. This will failover the active policy server based on a timeout value for DWR messages.
- You can configure Diameter STR message time-outs separately from all other Diameter messages by setting the **str-retry=<timeout number>** option in the external policy server configuration element. This will failover the active policy server based on a unique timeout value for STR messages.

If the Net-Net SBC sends an AAR/STR message to the active policy server and then switches to a different policy server, any new Diameter messages related to that session are sent to the same policy server as long as it is not inactive. If that server becomes inactive, messages will be sent to the new policy server. The new policy server however will not recognize the sessionID and discard the request.

Maintenance and Troubleshooting

The **show policy-server** command displays information specific to each policy server configuration by name, including HA cluster priority and state. For example:

```
ACMEPACKET # show policy-server seagull
name = seagull
address = host6.ipv6.gy
-----
192.168.24.40:3869, priority = 0, state = active
-----
192.168.24.40:3868, priority = 1, state = standby
-----
192.168.24.40:3870, priority = 2, state = inactive
TCP socket failures = 0
DIAMETER failures = 0
-----
192.168.24.75:3868
-----
```

Diameter Connection

The Net-Net SBC supports Diameter (RFC 3588) connections to a Diameter server over TCP. The base Diameter protocol runs on TCP port 3868. Diameter-based CAC services are available from the media interfaces (NIUs) on the Net-Net SBC.

The Diameter connection is always initiated from the Net-Net SBC to the Diameter server. The Net-Net SBC begins the connection by sending a Capabilities-Exchange-Request (CER) to the server, which replies with Capabilities-Exchange-Answer (CEA) message.

Diameter Heartbeat

Device-Watchdog-Request (DWR) and Device-Watchdog-Answer (DWA) messages are used to detect transport failures at the application layer between the Net-Net SBC communicating with a policy server via Diameter. The

request/answer message pair forms a heartbeat mechanism that can alert the requesting side if the answering side is not reachable.

The Net-Net SBC always responds to a DWR by replying with a DWA message. In addition, the Net-Net SBC can be configured to initiate DWR messages toward a policy server or other Diameter-based network device.

You configure the **watchdog ka timer** with a timeout value that determines the number of seconds a DWA is expected in response to the Net-Net SBC sending a DWR.

If the Net-Net SD fails to receive a DWA response from a Policy Server within the configured interval, then the connection towards that Policy Server is considered failed and torn down. The Net-Net SBC attempts to recreate the TCP connection, followed by the recreating the DIAMETER connection by issuing a CEA toward the policy server.

Setting the watchdog ka timer parameter to 0 indicates that the Net-Net SBC waits for incoming DWRs, only.

Diameter Failures

During periods of application inactivity on the Diameter interface, Device-Watchdog-Request (DWR) and Answer (DWA) messages are exchanged between the client and server to provide an application-level heartbeat. DWRs may be sent toward the Net-Net SBC, which responds with a DWA message.

A Diameter connection failure is determined by noticing that the TCP socket has disappeared. If the Diameter connection fails, the Net-Net SBC tries to re-open the TCP socket and Diameter connection to the Diameter server at 30 second intervals. The Net-Net SBC increases its retry interval to 5 minutes, until a successful Diameter connection is made.

Application IDs and Modes

Diameter messages include an application ID to indicate the application and standards' body interface. The following table lists common Application-IDs for the corresponding standards' and applications. Application IDs must be provisioned manually.

Standards Reference Point			
RACF			
AVP	Gq 3GPP R6 29.209	Rx 3GPP TS 29.214	Rq ETSI 283 026
Application-ID	16777222	16777236	16777222

You also set the application mode to specify the interface more precisely. Doing so avoids the potential for collision between interface types that can occur when you only configure the application identifier. By setting both the application mode and application identifier for the interface, you configure the format for DIAMETER messages.

The following table describes the application mode settings.

Application Mode Type	Description
Rq	As the default mode for the interface, Rq is the Net-Net SBC's base RACF interface. Even when you leave the application mode set to none (default), the Net-Net SBC runs in Rq mode. The only exception to this rule is if you set the application identifier to 16777236 and leave the application mode set to none; in this instance, the interface runs in Rx mode.
Rx	The interface runs in Rx mode when you either: - Set the application mode to Rx and the application identifier to 16777236 - Leave the application mode set to none, and set the application identifier to 16777236
Gq	The interface runs in Gq mode when you set the application mode to Gq. Note that the application identifier 1677722 is no longer unique, but applies both to Rq and Gq interface modes.
none	The interface runs in Rq mode when you do not configure an application identifier, or in Rx mode if you set the application identifier to 16777236.

Asynchronous SIP-Diameter Communication

The Net-Net SBC's Diameter-based external policy server support offers an asynchronous mode which can address the following two situations:

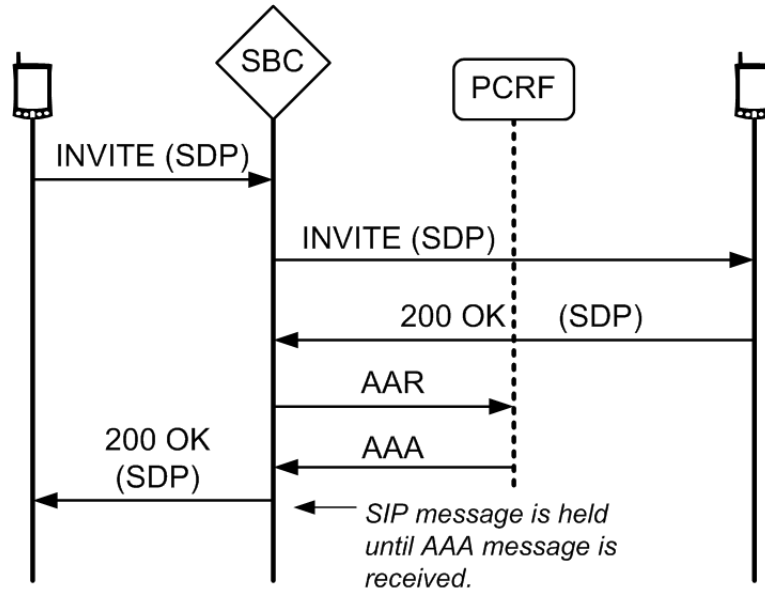
One of the fundamental behaviors in the Net-Net SBC's Diameter model relies on the External Policy Server making an authorization decision which is then communicated to the Net-Net SBC. Part of the call authorization sequence of events involves the Net-Net SBC waiting for an external policy server's response before the SBC can completely set up, modify, or update the call. The long pause that the endpoints experience, while the Net-Net SBC holds up SIP flows waiting for the external policy server's response, may lead to unnecessary call failure situations.

In some Diameter-based external policy server deployments, the media traverses a CMTS at the edge of the network; the CMTS gates may be established by a Policy Server to dynamically enable QoS to/from a UE toward another UE. If no gate is established then the media should still traverse the CMTS and be admitted to the network with a "best-effort" network path.

Synchronous Model

The Net-Net SBC's default external policy server model may be referred to as the synchronous model; The Net-Net SBC sends a Policy Server request based on a

SIP/SDP trigger point, the SIP signaling is held until a response is returned from the Policy Server. For example:



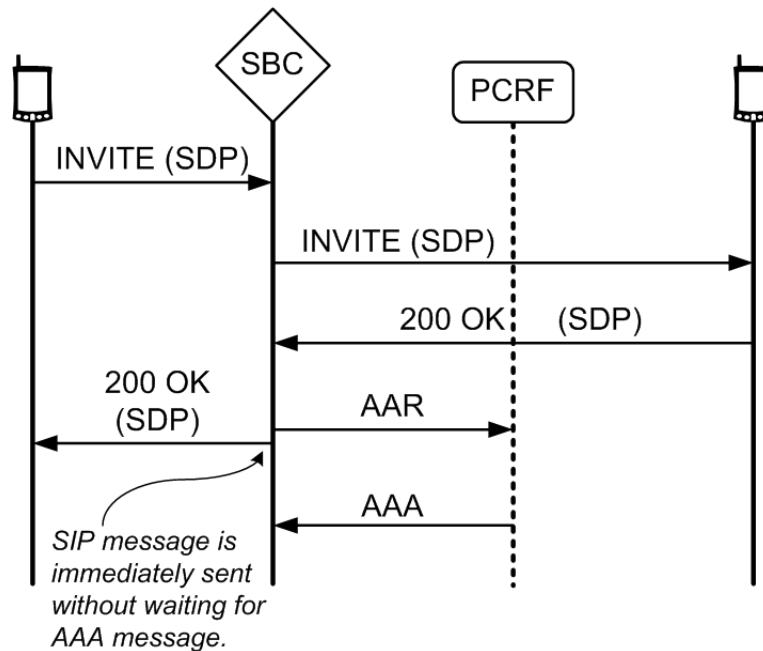
Note: In the above diagram, the **reserve incomplete** parameter has been set to **disabled**.

Asynchronous Model

In the asynchronous model the request that the Net-Net SBC's sends to the Policy Server flows in an asynchronous state with respect to SIP messaging. Thus the Net-Net SBC allows the SIP session to proceed naturally, and does not pause for outstanding Policy Server answers to be received.

It is crucial to understand that the Policy Server's ultimate success or failure answer, or timeout, as related to a SIP session does not impact the establishment of the SIP session itself.

The following diagram represents the asynchronous model where SIP sessions always succeed. SIP signaling is never delayed and as a result, call set-up time is decreased.



Note: In the above diagram, the **reserve incomplete** parameter has been set to **disabled**.

Caution

Acme Packet recommends that for each Net-Net SBC, the same model (Synchronous / Asynchronous) is used for all external policy servers configuration instances. Failure to follow this guideline may result in complex interactions from a timing perspective which may lead to a negative end user experience.

Serialized Diameter Messaging

Access networks with complex policy server structures might introduce the possibility of un-ordering Diameter requests into a given CMTS, even if Diameter message delivery was correctly ordered on the TCP connection between the Net-Net SBC and a lower-tiered external policy server.

The Net-Net SBC can prevent the external policy server from receiving out-of-order messages at the application layer by serializing them. The Net-Net SBC serializes Diameter messages by ensuring a Diameter request for one sessionID is not sent until an answer is received from the previous request for the same sessionID. The Net-Net SBC applies this constraint while waiting for Diameter response or when considering a Diameter request timeout (15 seconds).

Serialized Diameter messaging is always enforced for Diameter-based External Policy Server communication.

High Availability Considerations

Upon a failover, the Net-Net SBC may not guarantee the synchronicity of Diameter messages for active sessions. For example, if the active Net-Net SBC sent an AAR and an AAA is yet to be received, upon failover the newly active SBC may send a 2nd AAR or an STR without receipt of the initial AAA. This creates an unavoidable race condition. Such rare cases must be handled appropriately within the Quality of Service subsystem (Policy Servers and CMTSs).

Bandwidth-Based Call Admission Control

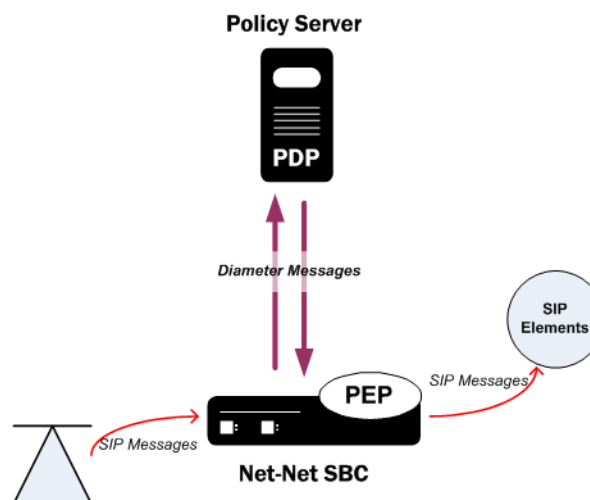
As the Net-Net SBC proxies and forwards calls, caller media information is known before the INVITE reaches the callee. The Net-Net SBC, acting as a PEP, requests a specific amount of bandwidth for a call, and the RACF can reserve this amount of bandwidth for a call before the called phone rings. A call's required bandwidth can also be reserved by network devices along the path from the caller to callee if the QoS admission criteria is pushed from the RACF to other edge nodes (PEPs) such as routers, along this path to the callee.

Implementation Features

Bandwidth-based CAC is performed according to the following typical scenario. When the Net-Net SBC, known as the Policy Enforcement Point (PEP), receives a SIP INVITE, it sends a Diameter Authentication Authorization Request (AAR) message to the Policy Decision Point (PDP) or Resource and Admission Control Function (RACF). The Net-Net SBC does not forward the INVITE to its destination at this point.

The AAR message includes call identification information and the SDP-based bandwidth requirements for the call. The RACF responds with a Diameter Authentication Authorization Answer (AAA) message to either the install or remove the call. An install command directs the Net-Net SBC to forward the INVITE to the next SIP device. A remove command directs the Net-Net SBC send a SIP 503 Service Unavailable message sent back to the UA and reject the call.

When the RACF is unreachable, incoming calls are rejected by default with a 503 message, as bandwidth can not be reserved. It is possible to configure the Net-Net SBC to allow all calls when the RACF is unreachable if this is the desired behavior.

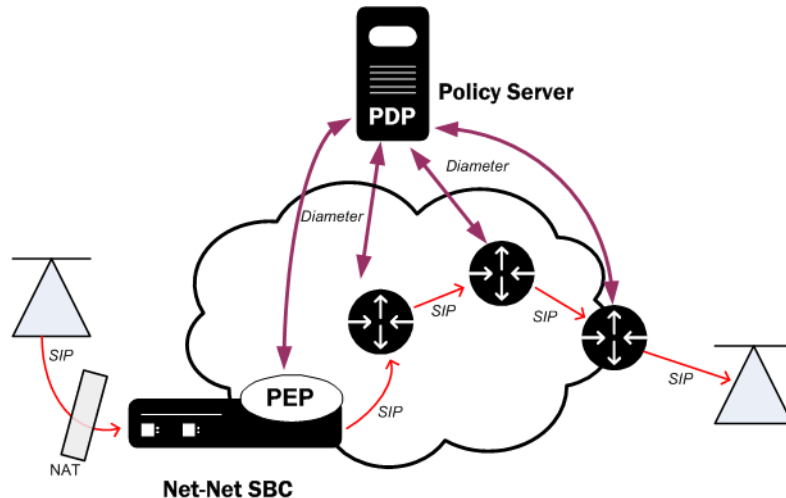


The Net-Net SBC can be configured so that both sides of a call, based on realm, are subject to bandwidth enforcement. Each flow is treated as a unique call/event, because from a media and signaling perspective, they are distinct. As the Net-Net SBC functions as one side of a call, its IP address is inserted into the AAR message

regardless of whether it is the calling or called party. This allows for the Diameter install or remove decision to be made before the Net-Net SBC receives the 200 OK response, and before ringing the far-end phone. Only one external policy server can be used within a single realm.

When a call ends, either with the expected SIP BYE or CANCEL methods, or due to other error conditions, the Net-Net SBC alerts the RACF by sending it a Diameter Session Termination Request (STR) message. All ended calls must be deleted from the RACF in order to accurately track used and available bandwidth.

The RACF can apply its hosted policies for calls originating at SIP UAs located behind NATs. This is a standard part of the Net-Net SBC's ability to provide seamless HNT.



Bandwidth Negotiation

Because the decision whether to admit or reject a call is made before the INVITE is forwarded to the called party, some information is not available to the PDP at the initial request. The final IP Address, UDP port number, that transport the RTP flow, and the codec used are not known by the Net-Net SBC until the called party responds with its own SDP information (either in the 180 or 200 response).

The Net-Net SBC examines the Session Description Protocol (SDP) value in the body of the SIP INVITE to determine what codecs are available for the call. If the INVITE specifies more than one codec, the Net-Net SBC bases its request to the RACF on the most bandwidth-hungry codec to ensure that all bandwidth requests will succeed or fail on the first attempt.

Note: The amount of bandwidth requested depends is based on the default and configured media profiles.

If the call is admitted, and when the called party returns finalized SDP information, the Net-Net SBC modifies the original reservation with the chosen codec's bandwidth requirements. This ensures the RACF has current and accurate information with which to make policy decisions.

Session Lifetime

When receiving a successful Diameter response message for bandwidth from the RACF, a session lifetime timer may be included in the message. If included, this timer states how long the session can last. If the session continues past 3/4 of session

lifetime, the Net-Net SBC sends another bandwidth request for that session to ultimately refresh the lifetime timer. If the RACF grants this bandwidth request, the Net-Net SBC continues to allow the session to proceed uninterrupted. If a lifetime timer for a session is not returned to the Net-Net SBC by the RACF, the Net-Net SBC assumes the session can last forever and never issues a refresh in this manner.

Destination Realms

The Diameter RACF interface can change the format of the payload string in the Destination-Realm AVP for any Diameter message it originates and sends to an external server. The payload field for this AVP can be constructed in any the following formats:

Format	Description
<user>@<realm>	<user>—IP address of the endpoint initiating the call with the Net-Net SBC <realm>—Name of the realm on which the Net-Net SBC received the INVITE from a user
<user>	<user>—IP address of the endpoint initiating the call with the Net-Net SBC
<realm>	<realm>—Name of the realm on which the Net-Net SBC received the INVITE from a user

When the Diameter RACF interface sends any message with the Destination-Realm AVP, it determines from the external policy server configuration how to construct the payload string for this AVP.

You can set the format to use in the **dest-realm-format** parameter in the external policy server configuration. The parameter can be set to any value in the table above and defaults to <user>@<realm>. By treating the format this way, the policy server and the Net-Net SBC can easily communicate this value; if sent to the policy server in any AVP, the policy server can simply return the full value.

Origination and Host Realms

The Diameter RACF interface can change the suffix for Origin-Realm and Origin-Host AVPs that have a payload string constructed as a domain name.

You can set the suffix you want appended to payload strings using the **domain name suffix** parameter in the external policy server configuration. This parameter can be set to any string (default is .com), and the Net-Net SBC automatically adds a dot (.) to the front of this entry if you do not include one. The policy server and the Net-Net SBC can easily communicate this value; if sent to the policy server in any AVP, the policy server can simply return the full value.

ACLI Instructions and Examples

In this example, you will configure additions to the realm configuration and the new external bandwidth manager configuration. You must also configure media profiles to accept bandwidth policing parameters.

Configuring the External Bandwidth Manager

To configure the external bandwidth manager:

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


```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press <Enter> to access the media-related configurations.

```
ACMEPACKET(configure)# media-manager
```
3. Type **ext-policy-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```
4. **name**—Enter the name for this external bandwidth manager instance. This parameter is used to identify the PDP that will be used in each Realm configuration.
5. **state**—Set the state of this ext-policy-server configuration to **enabled** to run this policy server. The default value is **enabled**. The valid values are:
 - enabled | disabled
6. **operation-type**—Enter **bandwidth-mgmt** for this external policy server configuration element to perform bandwidth management. The default value is **disabled**. The valid values are:
 - **disabled**—This parameter is disabled
 - **bandwidth-mgmt**—Net-Net SBC communicates with a CLF to obtain location string
7. **protocol**—Enter **Diameter** to support Diameter-based CAC.
8. **address**—Enter the IP address or FQDN of the external policy server. To use external policy server redundancy, you must enter the address as an FQDN.
9. **port**—Enter the port number the diameter connection connects to on the PDP. If you configure this parameter, it will override the port returned in a DNS reply. The valid range is:
 - Minimum—0
 - Maximum—65535
10. **realm**—Enter the name of the Realm in which this Net-Net SBC defines the PDP to exist. This is NOT necessarily the Realm where the Net-Net SBC performs admission control.
11. **permit-conn-down**—Enter **enabled** if this external policy server configuration can permit new calls into the network when the policy server connection is down. The default value is **disabled**. The valid values are:
 - enabled | disabled
12. **permit-on-reject**—Change this parameter to enabled if you want the Net-Net SBC to forward the session on at a “best-effort”. Leave this parameter set to disabled (default), if you want the Net-Net SBC to deny the session on attempts to revert to the previously-requested bandwidth. The valid values are:
 - enabled | disabled
13. **disconnect-on-timeout**—Leave this parameter set to **enabled** (default) so the Net-Net SBC maintains its TCP connection to the external policy server regardless of the upstream issues between policy servers (PS) and cable modem termination systems (CMTSs). When you disable this setting, the Net-Net SBC sends Gate-Set and Gate-Delete messages in response to the PS’s timeouts and guards against impact to the TCP connection between the Net-Net SBC and the PS. The valid values are:
 - enabled | disabled

14. **asynchronous-mode**—Leave this parameter at its default, **disabled** to run this external policy server in synchronous mode.
15. **product-name**—Enter text string that describes the vendor-assigned name for the RACF. This parameter is required.
16. **application-mode**—Enter the type of interface you want to use. Your choices are: **Rx**. For details about each type, refer to this chapter's [Application IDs and Modes \(806\)](#).
17. **application-id**—Enter a numeric application ID that describes the interface used to communicate with the RACF. Refer to the Application ID table on page [Application IDs and Modes \(806\)](#). The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
18. **framed-ip-addr-encoding**—Enter the format of the Frame-IP-Address (AVP 8) value in Diameter messages. The default value is **octet-string**. The valid values are:
 - **ascii-string**—Example: 192.168.10.1
 - **octet-string**—Example: 0xC0A80A01
19. **ingress-realm-location**—Set this parameter to configure the child realm, or its parent, for the Address-Realm in the Globally-Unique-Address AVP in DIAMETER UDR messages that the Net-Net SBC sends to the policy server. The valid values are:
 - **realm-in** (default)—The Net-Net SBC will use the same realm on which the REGISTRATION request arrived
 - **sip-interface**—The Net-Net SBC will use the realm associated with the SIP interface on which the REGISTRATION request arrived
20. **reserve-incomplete**—Set this parameter to **enabled** when communicating with a PDP. The parameter allows the SBC to make admission requests before learning all the details of the flows and devices (e.g., not knowing the final UDP port numbers for the RTP media streams until after the RTP has begun). The default value is **enabled**. The valid values are:
 - enabled | disabled | orig-realm-only
21. **watchdog-ka-timer**—Enter the interval in seconds of Net-Net SBC-initiated watchdog/keep-alive messages. Valid values range between 0 and 65535 seconds.
22. **dest-realm-format**—Enter the format you want to use for the Destination-Realm AVP. The default value is **user_with_realm**. The valid values are:
 - user_with_realm | user_only | realm_only
23. **domain-name-suffix**—Enter the suffix you want to use for Origin-Realm and Origin-Host AVPs that have a payload string constructed as a domain name. Your value can be any string, to which the Net-Net SBC will prepend with a dot if you do not include one. The default value is **.com**.
24. **gate-spec-mask**—With this parameter, you can configure the Net-Net SBC to use a mask comprised entirely of zeros (0). The default value is 255. This parameter sets the value to use for the COPs pkt-mm-3 interface. This interface maintains a persistent TCP connection to the external policy server, even without responses to requests for bandwidth. This permits calls to traverse the Net-Net SBC even though the external policy server either fails to respond, or rejects the session.

- Values Min: 0 / Max: 255
25. **max-timeouts**—Set the number of request timeouts before the Net-Net SBC sets this external policy server to inactive. You can separately set the number of DWR timeouts that trigger a server to be inactive as an option. Valid values range from 0 - 200.
 26. **max-connections**—Set the number of servers to maintain in an external policy server cluster. Valid values range from 1 - 20.
 27. **srv-selection-strategy**—Leave this parameter at its default, *Failover* setting.
 28. Save your work using the ACLI **done** command.

Configuring a Realm for Diameter Support

To configure the realm configuration for Diameter support in a CAC scenario:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)#
```
4. Type **select** and the number of the pre-configured sip interface you want to configure.


```
ACMEPACKET(realm-config)# select 1
ACMEPACKET(realm-config)#
```
5. **mm-in-realm**—Set this parameter to **enabled** so that calls from devices in the same realm have their media flow through the Net-Net SBC to be subject to CAC. The default value is **disabled**. The valid values are:
 - enabled | disabled
6. **mm-in-network**—Set this parameter to **enabled** so that the Net-Net SBC will steer all media traveling between two endpoints located in different realms, but within the same network. If this field is set to disabled, then each endpoint will send its media directly to the other endpoint located in a different realm, but within the same network. The default value is **enabled**. The valid values are:
 - enabled | disabled
7. **ext-policy-server**—Enter the name of the external bandwidth manager configuration instance to be used for external CAC for this Realm.
8. Save your work using the ACLI **done** command.

Additional Diameter Features

The Proxy Bit

When a signaling protocol receives an event request, the Net-Net SBC must ensure that the external policy server on the other end has enough bandwidth to maintain the requested call. The SDP information from the signaling message is stripped and encoded into the Diameter Band Request to be forwarded onto the external policy server.

The proxy bit allows the Net-Net SBC to tell the external policy server whether it wants the main server to handle the Diameter message, or if it is okay to proxy it to another server on the network. A parameter in the **ext-policy-server** configuration element called **allow-srv-proxy** has been developed. When this parameter is enabled, the proxy bit is set and the external policy server must process this Diameter request. When the parameter is disabled, the Net-Net SBC gives the external policy server permission to proxy the request along.

If you do not use this feature, this external policy server either handles the Diameter message on its own or proxies it to another server, depending on how much traffic it is handling at the time. This is done without any input from the Net-Net SBC.

To configure the proxy bit:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **ext-policy-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```
4. **allow-srv-proxy**—Set to **enabled** in order to include the proxy bit in the header. The presence of the proxy bit allows the Net-Net SBC to tell the external policy server whether it wants the main server to handle the Diameter message, or if it is okay to proxy it to another server on the network (**disabled**). The default is **enabled**. The valid values are:
 - enabled | disabled
5. Save your work using the ACLI **done** command.

Experimental Result AVP

The Net-Net SBC can accept, parse, and process the Experimental-Result AVP (which indicates that an error has occurred) whether accompanied by the Result-Code AVP or not. When a RACF interface receives the Experimental-Result AVP, it maps returned and corresponding values to construct responses for the requesting signaling application (SIP).

How It Works: RACF Experimental Result Handling

When the Diameter RACF interface receives an Authentication-Authorization-Answer (AAA) message that contains the Experimental-Result AVP, it will parse that AVP. The Experimental-Result AVP is a grouped AVP that includes the Experimental-Result-Code AVP, which contains an enumerated value in its payload that specifies the status of the received AAA message—either success or failure:

- **Success**—In the case where the enumerated value indicates success, the Net-Net SBC's Diameter RACF internally notifies SIP signaling of its status.
- **Failure**—In the case where the enumerated value indicates failure (or non-success), the Diameter RACF interface communicates internally with the Net-Net SBC's SIP signaling application, which will then reject the request with a 503 Service Unavailable response.

Optional AVP Support

The Net-Net SBC can use optional and non-standard AVPs in DIAMETER messages, including the Transport-Class AVP. Remaining as compliant as possible with current standards, the Net-Net SBC supports non-standard AVPs to extend its base interface modes and thereby increase its operating flexibility.

You can configure one or more AVPs in the external policy server configuration using the options parameter. The Net-Net SBC needs to know in what messages each optional AVP will appear, where the AVP will be located, what AVP flags to set, vendor information, and what data comprises the AVP's payload. Then the Net-Net SBC can insert the optional AVP in outgoing messages and parse incoming messages for it.

Transport-Class AVP

When it receives a SIP INVITE triggering external bandwidth management, the Net-Net SBC performs SDP stripping and—through internal processes—selects an external bandwidth manager to use. If the options parameter in the selected external bandwidth manager is set to **transport-class**, the Net-Net SBC's DIAMETER RACF interface will issue authentication authorization requests (AARs) with the transport class AVP. The Net-Net SBC does not insert the transport-class AVP messages when the option is not configured.

The transport-class AVP will:

- Be identified with the AVP code 311
- Always have the vendor (V) bit set in the AVP flags
- Never have the mandatory (M) bit set in the AVP flags
- Have the Vendor-Id field set to 13019 (a value specified by ETSI TISPAN)
- Be formatted as an unsigned integer
- Reside in the Media-Component AVP, a grouped AVP

In addition, the transport-class AVP's payload field will be a 32-bit unsigned integer corresponding to a specific media type. The Net-Net SBC learns the specific media type from the m-line of the SDP it received. The following table shows how the Net-Net SBC evaluates the SDP's m= lines and concludes a service type classification.

Service Class Classification	SDP evaluation	Default transport-class value
video	At least 1 m=video line	1
audio	No m= video At least 1m=audio	0
application	No m=video, No m=audio At least 1 m=application	3
data	No m=video, No m=audio, No m=application At least 1 m=data	2
control	No m=video, No m=audio, No m=application, No m=data At least 1 m=control	4
image	No m=video, No m=audio, No m=application, No m=data, No m=control At least 1 m=image	10

Service Class Classification	SDP evaluation	Default transport-class value
text	No m=video, No m=audio, No m=application, No m=data, No m=control, No m=image At least 1 m=text	5
message	No m=video, No m=audio, No m=application, No m=data, No m=control, No m= image, No m=text At least 1 m=message	6

After the service class classification has been chosen, the Net-Net SBC inserts the corresponding default transport class value in the transport-class AVP to send to the RACF.

Overriding Transport Class AVP Value

You can override the transport class AVP value sent to the RACF in a Diameter message by configuring the **service class options** parameter. Custom service class option values are configured as a <service-class-option>=<user-entered-value> value pair. For example, to configure the Net-Net SBC to send sending the value 80 instead of the value 8 for a message service class classification, you would configure **message=80** in the service class options parameter.

Configuring Optional AVPs

You configure optional AVPs in the external bandwidth manager's **options** parameter. The **options** parameter can take one or more AVP values. The instructions below use the transport-class AVP as an example.

To set the transport-class AVP support for an external bandwidth manager:

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```
2. Type **media-manager** and press <Enter>.


```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **ext-policy-server** and press <Enter>.


```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **transport-class** with a “plus” sign in front of it. Then press <Enter>.


```
ACMEPACKET(ext-policy-server)# options +transport-class
```

If you type **options** and then the option value without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. **service-class-options**—Set this options parameter by typing **service-class-options**, a <space>, the service-class type, an “equal” sign, your custom value. Then press <Enter>. Enter multiple service class/value pairs separated by a comma. For example:


```
ACMEPACKET(ext-policy-server)# service-class-options
message=80,text=70
```
6. Save and activate your configuration.

Diameter Destination Realm AVP

As of S-D7.1.0, the Destination Realm AVP's value does not contain the realm of the incoming SIP message. Now, it contains the realm where the Policy Server resides as learned from the Origin-Realm AVP received in a CEA message from the Policy Server.

The Net-Net SBC can be configured with an option, *include-gua*, to retain the legacy behavior of sending an incoming SIP message's realm to a policy server. This is accomplished by sending the Globally Unique AVP in the AAR message to the policy server, by adding an option parameter to the external policy server configuration.

The following table summarizes the effect of provisioning the external policy server with the Globally Unique AVP option on each Diameter interface, as configured.

Diameter Interface	No <i>include-gua</i> option Configured (default)	Add <i>include-gua</i> option
Rq	AAR sends Globally Unique Address AVP	AAR sends Globally Unique Address AVP
Rx	AAR does not send Globally Unique Address AVP	AAR will contain Globally Unique Address AVP
Gq	AAR does not send Globally Unique Address AVP	AAR will contain Globally Unique Address AVP
E2	AAR sends Globally Unique Address AVP	AAR sends Globally Unique Address AVP

Legacy Destination-Realm AVP Behavior

The Diameter RACF interfaces can change the format of the payload string in the Destination-Realm AVP for any Diameter message it originates and sends to an external server. The payload field for this AVP can be constructed in any the following formats:

Format	Description
<user>@<realm>	<user>—IP address of the endpoint initiating the call with the Net-Net SBC <realm>—Name of the realm on which the Net-Net SBC received the INVITE from a user
<user>	<user>—IP address of the endpoint initiating the call with the Net-Net SBC
<realm>	<realm>—Name of the realm on which the Net-Net SBC received the INVITE from a user

When the Diameter RACF interface sends any message with the Destination-Realm AVP, it determines from the external policy server configuration how to construct the payload string for this AVP.

You can set the format to use in the **dest-realm-format** parameter in the external policy server configuration. The parameter can be set to any value in the table above and defaults to <user>@<realm>. By treating the format this way, the policy server and the Net-Net SBC can easily communicate this value; if sent to the policy server in any AVP, the policy server can simply return the full value.

Signaling Only P-CSCF

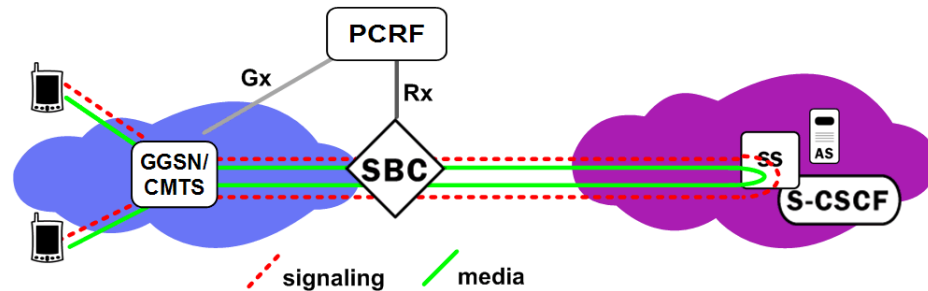
When configured as a P-CSCF, the Net-Net SBC can release media between two endpoints if the network topology allows it. Please refer to the following sections in the Realms and Nested Realms chapter to review the conditions and configurations required for media release:

- Media Release Across SIP Network Interfaces
- Media Release Behind the Same IP Address
- Media Release between Endpoints with the Same IP Address
- Media Release Behind the Same NAT IP Address

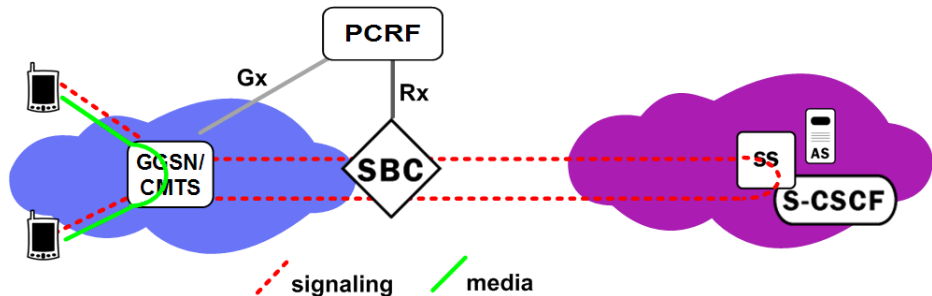
Media release with external policy server control is available when the Net-Net SBC, acting as a P-CSCF, connects via the Rx interface to a PCRF.

Media Path Deployment Models

After two UAs' session has been set up, the media follows the same path as the signaling, through the Net-Net SBC. This is the default managed media behavior.



In the released media scenario, as the call is set up, the Net-Net SBC determines if media release is possible. If it is, media is released between the two endpoints to take the most direct path. The Net-Net SBC manipulates the SDP so media is exchanged directly between the endpoints. While the signaling flows traverse the Net-Net SBC and other network elements, the media flows are routed or forwarded between the two endpoints.



Managed Media Advantages

The following applications are possible when the Net-Net SBC manages media:

- The Net-Net SBC can monitor media by maintaining flow guard timers. It also has tools to enforce bandwidth constraints on media streams.
- The Net-Net SBC can perform topology hiding by obfuscating core IP addresses between UAs.
- The Net-Net SBC can protect against signaling and bandwidth-based attacks.
- The Net-Net SBC can perform lawful interception on both signaling and media.

Released Media Characteristics

The following applications are possible when the Net-Net SBC releases media between UAs:

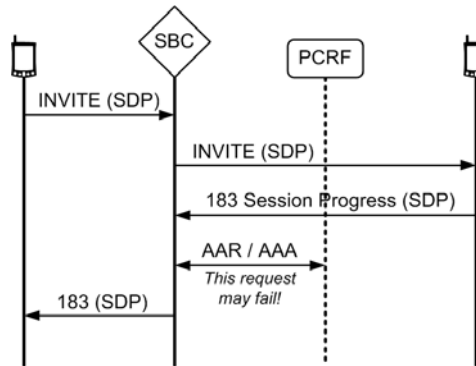
- When deployments include a MRFP and/or MGW, they need to monitor media flows and have access to endpoints' IP addresses; this is only possible with released media.
- The Net-Net SBC can protect against signaling-based DoS attacks.
- The Net-Net SBC can perform lawful interception only on signaling flows.

Bandwidth Request Models

After the Net-Net SBC identifies that a call can have its media released between endpoints, it refers to configuration for how to request bandwidth from the external policy server when setting up the call. The Net-Net SBC can use one of three models for requesting bandwidth and authorization from the external policy server.

Bandwidth Request Upon SDP Answer

In this model, fully specified policy server requests occur: the AAR message contains all final media and signaling information about the call, as gleaned from the SDP answer. Resources are requested and confirmed by the policy server after the SDP negotiation succeeds. In some cases, the callee UA may ring and subsequently be disconnected. The following is a simplified call flow.

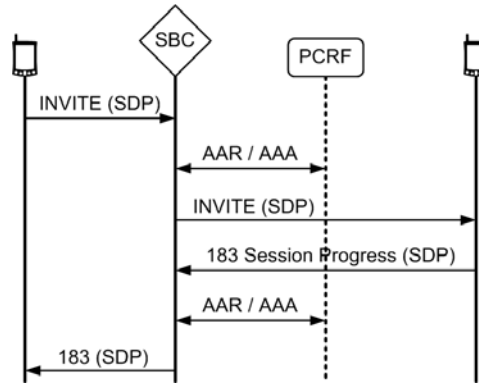


This is the default behavior of the Net-Net SBC and occurs when the **reserve incomplete** parameter is left to its default, **disabled**.

Bandwidth request Upon SDP Offer and Answer

In this model, partially specified policy server requests occur as the call is set up. When preliminary information is sent to the policy server for authorization, the AAR includes the Service-Info-Status AVP (code 527) with a value of 1 that indicates further negotiation will occur. Thus, resources are requested and confirmed by the policy server before ringing the callee, then once again after the SDP negotiation

succeeds. Reserved resources may be modified based on SDP answer depending on selected codec. The following is a simplified call flow.

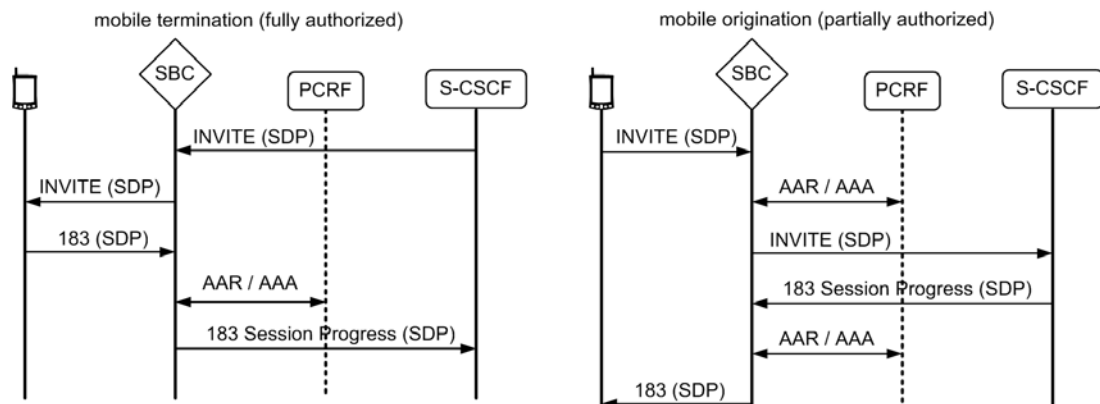


This behavior occurs when the **reserve incomplete** parameter is set to **enabled**.

Hybrid Bandwidth Request Model

This model, a hybrid of the first two, is useful in mobile scenarios. Some PCRF/PDF applications require the UA's location in order to successfully respond to an AAR query. When a UA has been idle for an extended period of time, its location needs to be refreshed by a page to determine the servicing ASN. This page typically takes the form of an L3 packet. As the phone's location has been refreshed, the Net-Net SBC can forward the INVITE and make a fully qualified AAR request that includes the location; none of the information in the AAR is provisional.

At the same time, when the call originates in the access network, the endpoint's location is known to the network upon receipt of the INVITE - the endpoint left its idle state. Since all expected information is known, the Net-Net SBC can make a provisional AAR query to the external policy server like in the Bandwidth request Upon SDP Offer and Answer.



This behavior occurs when the **reserve incomplete** parameter is set to **orig realm only**.

ACLI Instructions and Examples

The Net-Net SBC must first allow media release, which is configured by realm. Ensure that you have at least one of the following realm parameters configured to enabled:

- mm-in-realm
- mm-in-network
- mm-in-system
- mm-same-ip

There is no external policy server configuration associated with this feature.

To configure the external policy server's media release behavior:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **ext-policy-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```

Type **select** <Enter> to choose and configure an existing external policy server configuration element.
4. **reserve-incomplete**—Set this parameter to one of the following values for the stated use:
 - **disabled**—bandwidth request upon SDP answer
 - **enabled**—bandwidth request upon SDP offer and answer
 - **orig-realm-only**—hybrid model of using bandwidth requests on SDP answer and SDP offer and answer depending on the call's origination from core or access realm.
5. Save and activate your configuration.

Multi-tiered Policy Server Support

When accepting a call in a 2 tiered policy server environment, the Net-Net SBC queries the Tier-1 policy server, running in stateless mode, for authorization. The Tier-1 PS, identifies an appropriate Tier-2 policy server for every incoming DIAMETER session. Once a Tier-2 PS is chosen to control a given session-ID, all corresponding transactions for that session are sent toward the chosen Tier-2 policy server. The mapping of session-ID to Tier-2 PS is performed by either the Tier-1 PS or the P-CSCF. Using a stateless Tier-1 PS in this capacity increases overall performance.

To act in the P-CSCF role with a stateless Tier-1 policy server, the Net-Net SBC recalls the Origin-Host-AVP value returned in the initial AAA response to the Diameter session establishment query. The Origin-Host-AVP value is then inserted into the Destination-Host-AVP in all subsequent Diameter messages (AAR, STR) corresponding to that session-ID.

Messaging Details

The Net-Net SBC's normal behavior when communicating with a Diameter based policy server is to use the received STR message's Origin-Host AVP, as returned in the CEA, populating an outgoing Destination-Host AVP. Enabling the **cache dest host** parameter inserts the Origin-Host-AVP value from the first AAA message into

the Destination-Host-AVP in all subsequent Diameter messages (AAR, STR) corresponding to that session-ID.

If the Origin-Host AVP value, as received by the Net-Net SBC, changes mid session, all subsequent Destination-Host-AVP values are updated with the change.

ACLI Configuration

To configure Multi-tiered policy server support:

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


```
ACMEPACKET# configure terminal
```
2. Type **media-manager** and press <Enter> to access the media-related configurations.


```
ACMEPACKET(configure)# media-manager
```
3. Type **ext-policy-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```
4. **cache-dest-host**—Set this parameter to **enabled** for the Net-Net SBC to participate as a P-CSCF in a multi-tiered policy server environment.
5. Save and activate your configuration.

HDR Support for External Policy Servers

Historical data recording (HDR) support includes data for External Policy Servers statistics using Diameter (Diameter messaging statistics for external policy servers). External Policy Server HDR statistics match the statistics available at the ACLI when executing the **show policy servers** command. See the Net-Net 9000 Maintenance and Troubleshooting Guide, System Management chapter for more information.

Overview

Acme Packet's Net-Net 9200 currently provides intra-system high availability (HA) support. The Net-Net 9200 monitors itself for internal process and network health. When it detects internal faults, the card where the fault was found transitions to standby state, while the standby card transitions to active state. This process happens without any user-perceived call loss.

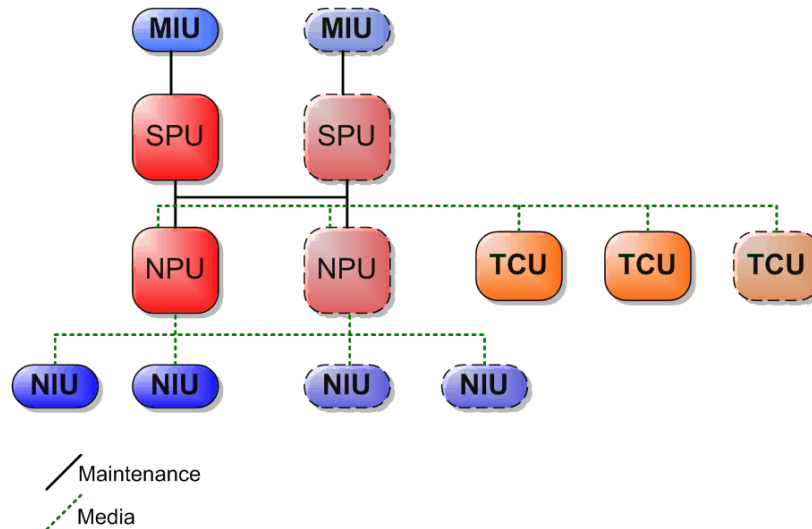
Intra-System HA Architecture

The Net-Net 9200 provides card-level component redundancy within the system chassis. Each active SPU, NPU NIU, and MIU has a backup card that provides 1:1 redundancy. This means that for one active device, there is one available backup device. The three TCUs can be configured for either 2:1 or 1:1 redundancy. This means that either of the two TCUs can fail over to the third redundant TCU, or that the Net-Net 9200 can support just one active and one standby TCU.

Each redundant card operates in hot-standby mode, capable of immediately assuming the active role. On a component failure, established, stable calls are maintained through the system. Call signaling and media flows proceed undisturbed through a hitless failover between the active and standby units; failover times occur in as little as 50ms.

As shown in the following diagram, each functional layer has one or more redundant components, and connects to multiple components in the adjacent layer so that there are no single points of failure. For example, when an NPU card fails, the standby NPU assumes the active role and communicates with the active SPU, TCUs, and NIUs.

Net-Net 9200 Block Diagram



HA Configuration and Management

The Net-Net 9200 continuously monitors itself to determine when a switchover should occur. A system of health scores is used to assign a severity value for system-affecting events. Switchovers occur when a minimum health score threshold is crossed, except when the standby component has a lower existing health score. The factors and thresholds have default values or may be provisioned by the system administrator.

A switchover triggers alarms, writes entries in logs, and causes SNMP traps to be issued. All notifications of these events are viewable through standard management means. In addition, LEDs on each feature card indicate whether it is active or standby.

Feature and Interface Card Redundancy

Feature and interface cards connect to each other through redundant communication paths within the Net-Net 9200 chassis. Multiple internal midplane connections provide this redundancy. The midplane is a passive electrical interconnect. No active components reside on the midplane board.

Signaling Processor Redundancy

Two SPUs provide hardware redundancy for VoIP signaling processing and general system functionality. Each SPU operates in either active mode or standby mode. When the standby SPU becomes active, the active NPU, NIUs, and TCUs all begin communication with the newly active SPU. After recovery, SPUs do not revert to their previous state until another failure or a forced switchover. Since MIUs are paired with SPUs, the MIU switches over when its SPU does.

Network Processing Redundancy

Two NPUs provide network processing and forwarding redundancy on the Net-Net 9200. Each NPU operates in either active mode or standby mode. When the standby NPU becomes active, the active SPU, NIUs, and TCUs all begin communication with the newly active NPU. When a standby NPU switches to active state, no other components explicitly switchover as a result of the NPU's switchover. After recovery, NPUs do not revert to their previous state, until another failure forces switchover.

Both NPUs' hardware components, such as the network processors, traffic managers, and CAMs are fully synchronized at all times. This enables an NPU switchover to occur rapidly without the need for re-populating the newly active NPUs with call state information.

Transcoding Redundancy

The Net-Net 9200 provides flexible transcoding redundancy with its three available TCUs. The TCUs by default use a 2:1 protection mechanism. This means that 2 active TCUs are backed up by a common standby TCU. Two TCUs operate in active mode and the third TCU operates in standby mode. When the standby TCU switches to active state, the active NPU communicates with the newly active TCU. When a standby TCU switches to active state, no other components explicitly switchover as a result of the TCU's switchover. After recovery, TCUs do not revert to their previous state until another failure or forced switchover.

The hardware components on the TCUs are fully synchronized at all times. Therefore, a TCU switchover occurs with no transcoding service loss for established calls and with no user-perceived impact.

Network Interface Redundancy

The Net-Net 9200 supports full media interface redundancy across its 4 NIUs. Each NIU operates in active mode or standby mode. Interfaces on an NIU will be either all active or all standby at once. All interfaces, whether active or standby, remain link up as they connect to other Ethernet ports, unless there is a link failure. NIU configuration can be either one active and one standby NIUs, two active and two standby NIUs, or two active no standby NIUs. After recovery, NIUs do not revert to their previous state, until another failure forces switchover.

Each active NIU is matched with a standby. Slot 0's redundant NIU rests in slot 2. Slot 1's redundant NIU rests in slot 3. The matching pair of active and standby NIUs must be the same physical interface type (quad copper or quad SFP). This is necessary because the physical cabling of active and backup interfaces must be identical, as well as common logical VLANs and virtual IP Addresses.

When cabling redundant NIUs, the active and standby interface pairs must exist in the same logical VLAN; they do not need to exist on the same physical switch.

Other than the [Gateway Link Failure Detection and Recovery](#) configuration, there is no explicit procedure to enable redundancy on an NIU.

Management Interface Redundancy

Management interface redundancy is provided by standby MIUs. Each MIU operates either in active mode or standby mode. Each SPU is matched to a corresponding MIU. When the standby SPU assumes the active role, the MIU paired with that SPU becomes the active too. This is unlike other Net-Net 9200 redundancy behaviors.

The console interface on each MIU is available at all times, regardless of the card's active or standby state. Management Ethernet interfaces use the gateway link failure and polling feature to continue using the same management IP addresses from a newly active MIU after failover. Thus, system administrators can always connect to

the Net-Net 9200's management IP addresses, or connect to the standby MIU using its console port. After recovery, MIUs do not revert to their previous state until another SPU or MIU failure forces a switchover.

Each MIU is paired with an SPU; the SPU in front-panel slot 0 uses the MIU in rear-panel slot 0, the SPU in front-panel slot 1 uses the MIU in rear-panel slot 1. After an SPU failover, the MIU paired with the newly active SPU becomes active. This arrangement keeps the unique management IP addresses logically paired with their respective SPU. Note that a failure of one or both MIUs is not service-impacting, and does not affect system availability.

You can also configure link failure detection for each MIU. When configured, the Net-Net 9200 pings the configured IP address; if there is no answer, then it triggers a link down alarm.

To configure MIU link failure detection:

1. In Superuser mode, type **configure terminal**.

```
ACMEPACKET# conf t
ACMEPACKET(configure)#
```
2. Access the system path for configuration.

```
ACMEPACKET(configure)# system
```
3. Access the system configuration. The system prompt changes to let you know that you can configure individual parameters.

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```
4. **miu0-monitor-ip-address**—Enter the remote IP address that you want to use for link failure detection for MIU 0.
5. **miu1-monitor-ip-address**—Enter the remote IP address that you want to use for link failure detection for MIU 1.

Health Scores

The Net-Net 9200 uses a system of health scores to determine when to switch between redundant feature and interface cards. Health scores are based on a 100-point system. When a component functions properly, its health score is 100.

Generally, the redundant card with the higher health score is active, and the card with the lower health score is standby. However, configurable health score thresholds build some flexibility into using health scores to determine active and standby roles.

The **show health** command is used to view health scores of all components within the Net-Net 9200 chassis. This command also displays the current state of each card. For example:

```
ACMEPACKET# show redundancy all
```

Slot	Type	Health Score	Role
0	SPU0	100	ACTIVE
1	SPU1	100	STANDBY
2	NPU0	99	ACTIVE
3	NPU1	99	STANDBY
4	TCU0	100	ACTIVE

5	TCU1	100	STANDBY
6	TCU2	100	STANDBY
7	MIU0	100	ACTIVE
8	MIU1	100	STANDBY
9	PHY0	100	STANDBY
10	PHY1	100	ACTIVE
11	PHY2	100	ACTIVE
12	PHY3	100	STANDBY
13	FANCTRL 0	100	ACTIVE
14	FANCTRL 1	100	ACTIVE
15	POWER0	100	ACTIVE
16	POWER1	100	ACTIVE
17	POWER2	100	STANDBY
18	POWER3	100	STANDBY

Failovers

A failover occurs when a pair of active and standby cards within the Net-Net 9200 switch roles. There are two kinds failovers: automatic and manual. A manual failover is also known as a switchover because no component failure occurs.

Automatic Failovers

Automatic failovers are triggered without intervention on the system administrator's part. Net-Net 9200 cards failover automatically in the following circumstances:

- When the active card's health score of drops below the threshold you configure.

Manual Switchover

Manual failovers force a pair of redundant cards to trade roles. The active card becomes standby, and the standby card becomes active. In order to manually switch card roles, the following conditions must be met.

- The card to be switched over must be in one of the following states: active, standby, or becoming standby.
- A manual switchover to the active state is only allowed on a feature card in the standby or becoming standby state if it has achieved full media, signaling, and configuration synchronization.
- A manual switchover to the active state is only allowed on a card in the standby or becoming standby state if it has a health score above the value you configure for the threshold.

The **switchover** command causes the redundant card pair to switch roles. You supply the slot number of a card and the new role for it to assume. For example:

```
ACMEPACKET# switchover 0 standby
```

Card State Transitions

Feature cards transition through a series of states as they become active or become standby.

Note: Packet processing and transcoding only occur on an active NPU or TCU.

State	Description
Initial	When the Net-Net 9200 is booting.
Becoming Active	Card is transitioning to active state. Packets cannot be processed in this state.
Active	Card is active; it can handle all media, signaling, and system processing.
Becoming Standby	Card is transitioning to standby role.
Standby Timeout	Card times out while waiting to become standby.
Standby	Card is fully synchronized with active peer card, but in standby role.
Becoming OOS	Card is transitioning to Out of Service (OOS) role.
OOS Timeout	Card times out while waiting to become OOS.
OOS	Card is in OOS state.
Recovery	Card's transitional state before it reverts to Standby or OOS.

Gateway Link Failure Detection and Recovery

Net-Net 9200s can detect when network connections to gateways are lost and then failover to a secondary network connection. The Net-Net 9200 monitors its network connectivity to the next hop by sending periodic ARP requests; this is known as a heartbeat message. When the Net-Net 9200 sends an ARP request and the next hop replies with an ARP reply, the connection is considered up, and service proceeds normally.

Link Failure Detection

If the Net-Net 9200 does not receive the expected ARP reply to its heartbeat ARP request within the specified timeout period, the gateway is considered unreachable and a "gateway unreachable" network-level alarm is generated. The Net-Net 9200 considers the link failed, sets its interface card to standby, and fails over to the redundant card. The redundant card assumes the active role. The configured health score amount is subtracted from the interface card that experienced the gateway link failure.

NIU link detection and gateway polling is configured on a per-interface basis.

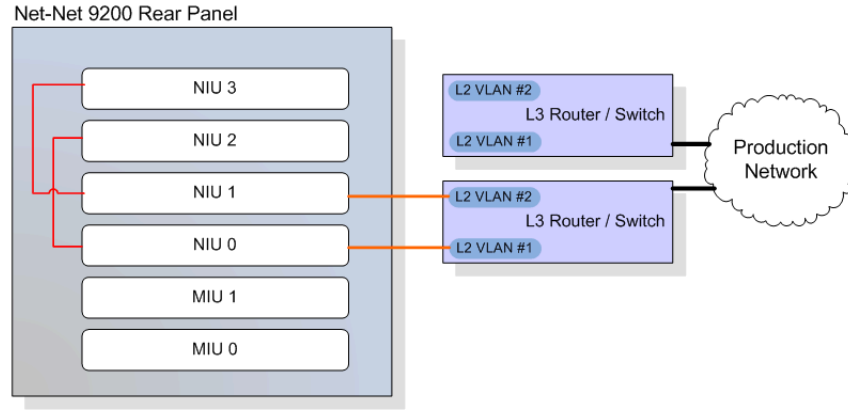
Warning: To improve the detection of link failures, the switchport connected to the NIU should have Spanning Tree disabled. Enabling Spanning Tree stops the switchport from forwarding frames for several seconds after a reset. This prevents the NIU from reaching the gateway and generates a "gateway unreachable" network-level alarm.

Link Failure Recovery

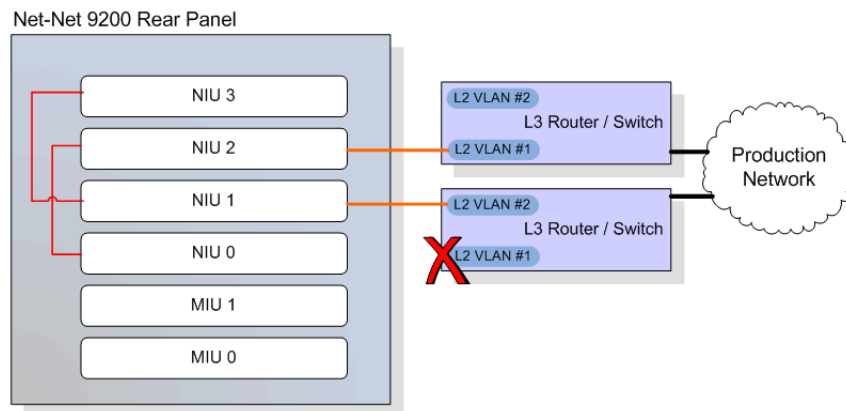
To produce seamless switchovers from one interface card to the other, the Net-Net 9200 uses a system of shared virtual MAC and virtual IP addresses in a similar way

to VRRP (virtual router redundancy protocol). When a network interface card fails over, the standby interface card sends out gratuitous ARP messages using a virtual MAC address, establishing that MAC on another physical port on the next-hop Ethernet switch or router. To the upstream device, the MAC and IP are still alive and unchanged so that existing network connections remain undisturbed.

The diagram below represents standard operation. NIU 0 and NIU 1 each connect to their respective VLAN on a switch.



In the following diagram, the Net-Net 9200 determines that NIU 0 should be taken out of service. NIU 2 becomes active on another router and establishes itself with the same IP addresses and MAC as NIU 0 had.



The network should be configured so that the change from one L3 interface to another has no effect on VoIP service.

ARP Polling Configuration

Three parameters determine how the Net-Net 9200 polls the next-hop for its Gateway Link Failure feature. The heartbeat parameter indicates the number of seconds the Net-Net 9200 waits between sending ARP requests to the gateway. The retry timeout is the number of seconds that must elapse between when the Net-Net 9200 sends an ARP request and does not get a response back from the gateway. When this time is exceeded, the Net-Net 9200 starts counting the successive failed ARP requests. The retry count is the number of successive failed ARP requests that must accumulate before the interface is declared out of service (and the interface fails over to the redundant device).

ACLI Instructions and Examples

To configure interface link detection and gateway polling for a single network interface in the ACLI:

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


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


```
ACMEPACKET(configure)# system
```
3. Type **network-interface** and press <Enter>.


```
ACMEPACKET(system)# network-interface
```
4. Type **gw-heartbeat** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.


```
ACMEPACKET(network-interface)# gw-heartbeat
ACMEPACKET(gw-heartbeat)#
```

From here, you can configure gateway heartbeat parameters for the network interface. To view all gateway heartbeat parameters, enter a ? at the system prompt.
5. **state**—Set this parameter to **enabled** to use the gateway heartbeat feature, or leave it **disabled** (default) to disable the feature.
6. **heartbeat**—Enter the number of seconds between heartbeats for this interface; this parameter defaults to 0. Heartbeats are sent at this interval as long as the interface is viable.
7. **retry-count**—Enter the number of heartbeat retries that you want sent to the front interface gateway before it is considered unreachable; this parameter defaults to 0.
8. **retry-timeout**—Enter the heartbeat retry time-out value in seconds; this parameter defaults to 1.

This parameter sets the amount of time between Net-Net 9200 ARP requests to establish interface gateway communication after an interface gateway failure.
9. **health-score**—Enter the amount to subtract from the Net-Net 9200's health score if an interface heartbeat fails; this parameter defaults to 0. If the value you set in the **retry-time-out** field is exceeded, this amount will be subtracted from the system's overall health score.

NPU Failure Detection and Recovery

The NPU runs two tests to monitor its system health. These tests are for NP Health and the Internal Path Test (IPT).

Network Processor Failure

The Net-Net 9200 monitors the NIU's Network Processors (NPs) for health. If they are operating at an unacceptable level, an alarm will be raised, the NIU will fail over to the healthy NIU, and the failed NIU will reset with a hard reboot (power cycle).

NPU IPT Failure

An NPU IPT runs continuously at fixed intervals during the normal operation of the Net-Net 9200 without any adverse affects to the system. The IPT validates that all the major components constituting the internal signaling path are operational. The IPT only sends an alarm if a gateway heartbeat alarm is active, at which time the Net-Net 9200 fails over the NPU to the healthy peer and resets the NPU where the IPT test failed. The IPT Failure alarm is triggered when this test fails.

Acronym List

General Use Acronyms

A

AAA—Administration, Authentication, and Authorization
ACD—Automatic Call Distribution
ACL—Access Control List
ACLI—Acme Command Line Interface
ACP—Acme Control Protocol
ACK—Acknowledgement (SIP)
ADSL—Asymmetric Digital Subscriber Line
AF—Access Function
AIN—Advanced Intelligent Network
ALG—Application Layer Gateway
ANI —Automatic Number Identification (ISDN)
ANSI—American National Standards Institution
AP—Application Protocol
API—Application Programming Interface
ANP—Access Point Name
APPN—Advanced Peer-to-Peer Networking
ARP—Address Resolution Protocol
ASCII—American Standard Code for Information Interchange
ASIC—Application-Specific Integrated Circuit
ASN—Abstract Syntax Notation
ASP—Application Service Provider, Active Server Pages, Adjunct Service Point
ASR —Access Service Request
ATM —Asynchronous Transfer Mode
ATX—Advanced Technology Extended

B

B2BUA—Back-to-Back User Agent
BBSRAM—Battery Backup Static Random Access Memory
BCID—Billing Correlation Identifier

BGP—Border Gateway Protocol
 BHCA—Busy Hour Call Attempts
 BIOS— Basic Input/Output System
 BITS—Building Integrated Timing Supply
 B-ICI—Broadband Inter-carrier Interface (ATM)
 B-ISUP—Broadband ISDN User Part
 BNF—(augmented) Backus-Naur Form
 BoS—Bottom of Stack
 bps—Bits per Second
 BRAS—Broadband Remote Access Server
 BRI—Basic Rate Interface (ISDN)
 BSP— Board Support Package
 BTU—British Thermal Units

C

CAC—Connection Administration Control (ATM)
 CALEA—Communications Assistance to Law Enforcement Agencies
 CAM—Content Addressable Memory
 CARP—Cache Array Routing Protocol (to replace ICP)
 CAS—Cordless Access Service
 CBR—Constant Bit Rate (ATM)
 CC —Country Code
 CCC—Call Content Connection
 CCCid—Call Content Connection Identifier
 CCM—Cisco Call Manager
 CDC—Call Data Connection
 CF—Collection Function
 CDPN—Called Party Number
 CDR—Call Detail Record
 CE—Conformité Européenne (The CE marking is a European proof of conformity and is also described as "passport" that allows manufacturers and exporters to circulate products freely within the EU.)
 CFM—Cubic Feet per Minute (fan speed)
 CFU—Call Forwarding Unconditional
 CGI—Common Gateway Interface
 CIC—Carrier Indicator Code/Carrier Identification Code
 CID—Caller Identification
 CISSP—Certified Information Systems Security Professional

Client-SI—Client-Server Information
 CLEC—Competitive Local Exchange Carrier
 CLI—Command-line Interface
 CMIP—Common Management Information Protocol
 CMIS—Common Management Information Service
 CMOT—Common Management Information Services Over TCP
 CMS—Call Management Server
 CNM—Customer Network Management
 CO—Connection Oriented
 CODEC—Coder/Decoder
 COPS—Common Open Policy Service
 CORBA—Common Object Request Broker Architecture
 CoS—Class of Service
 CP—Communications Processor
 CPL—Call Processing Language
 CPLD—Complex Programmable Logic Device
 CPM—Communications Processor Module
 CPU—Central Processing Unit
 CRA—Call Routing Apparatus
 CRLF—Carriage Return Line Feed
 CSA—Client Server Architecture
 CSPDN—Circuit Switched Public Data Network
 CSU—Channel Service Unit
 CT—Cordless Telephone
 CT-1—European analogue cordless telephone system
 CT-2—Second-generation cordless telephone, Digital
 CTI—Computer Telephony Integration
 CUG—Closed User Group

D

DA—Destination Address
 DAM—Data Access Method; Data Asset Management
 DDD—Direct Distance Dialing
 DDF—Digital Distribution Frame
 DECT—Digital European Cordless Telephone
 DF—Distribution Function
 DHCP—Dynamic Host Configuration Protocol

DIAMETER—a protocol for authentication, authorization, and accounting
 DiffServ—Differentiated Services
 DIMM—Dual In-line Memory Module
 DLCI—Data Link Connection Identifier
 DLSR—Delay Since Last Send Report
 DN—Directory Number
 DNS—Domain Name Server/Service
 DOM—Document Object Model
 DoS—Denial of Service
 DP—Destination Port
 DPCM—Differential Pulse Code Modulation
 DRAM—Dynamic Random Access Memory
 DS—Differentiated Services
 DSCP—DiffServ Codepoint
 DSL—Digital Subscriber Line
 DSLAM—Digital Subscriber Line Access Multiplexer
 DSP—Digital Signal Processing
 DSS—Digital Satellite System
 DSU—Digital Service Unit
 dTCP—Dynamic Transmission Control Protocol
 DTD—Document Type Definition
 dTLS—Dynamic Transport Layer Security
 DTMF—Dial Tone Multi-Frequency

E

ED—Ending Delimiter
 EEPROM—Electrically Erasable Programmable Read-Only Memory
 EFTPOS—Electronics Funds Transfer Point of Sale
 EGP—Exterior Gateway Protocol
 EMC—Electromagnetic Compatibility
 EMI —Electromagnetic Interference
 ENUM—Refers to the use of an E.164 number, in reverse, with domain notation (i.e., dotted).
 EPROM—Erasable Programmable Read-Only Memory
 ER—Edge Router
 ESD—Electrostatic Discharge
 ETSI—European Telecommunications Standards Institute

F

FCC—Federal Communications Commission
 FCP—Firewall Control Protocol
 FEC—Forward Equivalence Class
 FPGA—Field Programmable Gate Array
 FQDN—Fully Qualified Domain Name
 FSA—Foreign SIP Agent (ACME-specific term?)
 FTP—File Transfer Protocol
 FTR—Flow Transform Record

G

GA—Global Address
 GB—Gigabyte
 GBPS—Gigabits Per Second
 GigE—Gigabit Ethernet
 GMII—Gig Media Independent Interface
 GNU—GNUs not UNIX
 GOSIP—Government Open Systems Interconnection Profile
 GPS—Global Policy Server/Global Positioning System
 GSM—Global Systems for Mobile Communications
 GSTN—Global Switched Telephone Network

H

HA—High Availability (Acme Packet redundancy solution)
 HNT—Hosted NAT Traversal (Acme Packet)
 HTML—Hypertext Markup Language
 HTTP—Hypertext Transport Protocol

I

I/O—Input/Output
 IAD—Integrated Access Device
 IANA—Internet Assigned Numbers Authority
 IAP—Intercept Access Point
 ICE—In Circuit Emulator
 ICMP—Internet Control Message Protocol
 ICP—Internet Cache Protocol
 IDS—Intrusion Detection System
 IEC—International Electrotechnical Commission
 IEEE—Institute of Electrical and Electronics Engineers
 IESG—Internet Engineering Steering Group

IETF—Internet Engineering Task Force
 IGP—Interior Gateway Protocol
 IIS—Internet Information Server
 ILEC —Independent Local Exchange carrier (USA)
 IN—Intelligent Network
 IOS—Internetworking Operating System
 IP—Internet Protocol (IPv4, IPv6)
 IPC—Inter-process Communication
 IPDR—Internet Protocol Data Record
 IPsec—Internet Protocol Security
 IPtel—Internet Protocol Telephony
 IPv—Internet Protocol version
 ISDN—Integrated Services Digital Network
 ISO—International Organization of Standardization
 ISP—Internet Service Provider
 ITAD—Internet Telephony Administrative Domain
 ITSP—Internet Telephony Service Provider
 ITU—International Telecommunication Union
 ITU-T—International Telecommunication Union Telecommunication
 Standardization
 IVR—Interactive Voice Response
 IXC—Interexchange Carrier

J

JTAG—Joint Test Action Group
 JTAPI—Java telephony application programming interface

K

Kb—Kilobits
 KB—Kilobytes
 Kbps—Kilobits per second
 KTS—Key Telephone System

L

LA—Local Address
 LAES—Lawfully Authorized Electronic Surveillance
 LAN—Local Area Network
 LATA—Local Access and Transport Area
 LCD—Liquid Crystal Display

LDAP—Lightweight Direct(ory) Access Protocol
 LEA—Law Enforcement Agency
 LEAF—Law Enforcement Administrative Function
 LEC—Local Exchange Carrier
 LED—Light Emitting Diode
 LEM—Local Element Manager
 LEN—Local Exchange Node
 LI—Lawful Intercept
 LNP—Local Number Portability
 LOS—Loss of Signal
 LS—Location Server
 LSB—Least Significant Bit
 LSR—Label-switching router

M

MAC—Media Access Control
 MAN—Metropolitan Area Networks
 Mb—Megabits
 MB—Megabytes
 MBCD—Middlebox Control Daemon
 Mbone—Multicast Backbone
 MCU—Multi-party Conference Unit
 MD5—Message Digest 5 (hash function)
 MF—Media Firewall
 MG—Media Gateway
 MHz—Megahertz
 MIB—Management Information Base (TMN)
 MIB II—Management Information Base II
 MIBOC—Middlebox Control Protocol
 MIDCOM—Middle Box Communications
 MIME—Multipurpose Internet Mail Extension
 MoIP—Messaging over Internet Protocol
 MP—Main Processor
 μ P—Microprocessor (subsystem)
 MPLS—Multi-protocol Label Switching
 MR—Media Router
 MRCP—Media Router Control Protocol

MSB—Most Significant Bit
MTA—Message Transfer Agent
MTA—Multimedia Terminal Adapter
MTBF—Mean Time Between Failures
MTTR—Mean Time To Repair
MTU—Maximum Transmission Unit
MX—Mail Exchange

N

N-ACD—Network Automotive Call Distribution
NANP—North American Numbering Plan
NAPT—Network Address Port Translation
NAS—Network Access Security
NAT—Network Address Translation
Nco—Network Code of Practice
NCP—Network Control Point
NCS—Network Call Signaling
NEBS—Network Equipment - Building Systems/Standards
NIC—Network Interface Card
NMS—Network Management Station
NP—Network Processor
NSRG—Network Signaling Record Generator
NTE —Networking Terminating Equipment
NTP—Network Time Protocol
NTU—Networking Terminating Unit
NVRAM—Non-volatile Random Access Memory

O

OAM—Operation, Administration, and Maintenance
OC—Optical Carrier
OC-n—Optical Carrier transport
OCx—Optical Carrier level
OEI—Optical Electrical Interface
OEM—Original Equipment Manufacturer
OID—Object Identifier
ONP—Open Network Provision
OS—Operating System
OSI—Open System Interconnect(ion)

OSP—Open Settlement Protocol
 OSPF—Open Shortest Path First
 OSS—Operation Support Systems

P

PABX—Private Automatic Branch Exchange
 PACS—Personal Access Communications Systems
 PAT—Port Address Translation
 PBX—Private Branch Exchange
 PCB—Printed Circuit Board
 PCDATA—Parseable Data Characters
 PCI—Peripheral Component Interconnect
 PCMCIA—Personal Computer Memory Card International Association
 PCN—Personal Communications Network
 PCS—Personal Communications Services
 PDCS—Packet Cable Distributed Call Signaling
 PDH—Plesiochronous Digital Hierarchy
 PDN—Public Data Network
 PDN—Packet Data Network
 PDP—Policy Decision Point
 PDU—Protocol Data Unit (or Packet Data Unit)
 PEP—Policy Enforcement Point/Protocol Extensions Protocol
 Perl—Practical Extraction Report Language
 PHY—Physical Layer Device
 PIB—Policy Information Base
 PING—Packet Internet Groper
 PINT—PSTN and IP Internetworking
 PKI—Public Key Infrastructure
 PMC—PCI Mezzanine Card
 PNNI—Private Network Node Interface (ATM)
 PNO—Public Network Operator
 POP—Point of Presence
 POS—Packet Over SONET
 POTS—Plain Old Telephone Service
 PPP—Point-to-Point Protocol
 PROM—Programmable Read-Only Memory
 PSTN—Public Switched Telephone Network (Telecom Network)

PTE—Packet Transform Engine
PTO—Public Telecommunications Operator
PTT—Post, Telephone, and Telegraph
PWB—Printed Wiring Board

Q

QoS—Quality of Service
QSIG—Unified International Digital Corporate Network Signaling Standard

R

RADIUS—Remote Authentication Dial-in User Service
RAM—Random Access Memory
RARP—Reverse Address Resolution Protocol
RED—Random Early Discard
REN—Ringer Equivalent Number
RFC—Request for Comments
RIP—Routing Information Protocol
RISC —Reduced Instruction Set Chip
RMON—Remote (Network) Monitoring
ROM—Read-Only Memory
RPC—Remote Procedure Call
RR—Received Report
RS-232—Recommended Standard 232 (computer serial interface, IEEE)
RSA—Rivest, Shamir, & Adleman (public key encryption technology)
RSVP—Resource Reservation Protocol
RTCP—Real-time Control Protocol
RTP—Real-time Transport Protocol
RTP/AVP—Real-time Transport Protocol/Audio-Video Protocol
RTSP—Real-time Streaming Protocol
RTT—Round Trip Time

S

SA—Source Address
SA—Session Agent
SAG—Session Agent Group
SBC—Session Border Controller
SCE—Service Control Environment
SCP—Service Control Point
SCTP—Streaming Control Transmission Protocol

SDH—Synchronous Digital Hierarchy
 SDP—Session Description Protocol
 SDRAM—Synchronous Dynamic Random Access Memory
 SERDES—Serial De-serializer
 SIG—Special Interest Group
 SIM—Subscriber Identity Module
 SIMM—Single In-line Memory Module
 SIP—Session Initiation Protocol
 SLA—Service Level Agreement
 SME—Small to Medium Enterprise(s)
 SMIL—Synchronized Multimedia Integration Language
 SMP—Simple Management Protocol
 SMTP—Simple Mail Transfer Protocol
 SNMP—Simple Network Management Protocol
 SOCKS—SOCKeT server
 SONET—Synchronous Optical Network
 SP—Source Port
 SP—Service Provider
 SR —(Net-Net) Session Router
 SRAM—Static Random Access Memory
 SRS—Session Routing System
 SRV—Resource record for servers (DNS)
 SS7—Signaling System 7
 SSH—Secured Shell or Secure Socket Shell
 SSL—Secure Socket Layer
 SSP—Service Switching Point
 sTCP—Static Transmission Control Protocol
 STL—Standard Template Library
 sTLS—Static Transport Layer Security
 STP—Signal Transfer Point; Service Transfer Point
 SVC—Signaling Virtual Channel (ATM)
 SVC—Switched Virtual Circuit (Packet Switching)

T

TA—Terminal Adapter (ISDN)
 TAC—Terminal Access Control
 TACACS+—Terminal Access Controller Access Control System

TAPI—Telephony Application Program Interface
 TAXI—Transparent Asynchronous Transmitter/Receiver Interface
 TCB—Task Control Bar/Task Control Block
 TCI—Tag Control Identifier
 TCP—Transmission Control Protocol
 TCP/IP—Transmission Control Protocol /Internet Protocol
 TEN—Transit Exchange Node
 TFTP—Trivial File Transfer Protocol
 TLS—Transport Layer Security (same as SSL)
 TLV —Tag Length Value
 TM—Traffic Manager
 TMN—Telecommunications Management Network
 ToS—Type of Service
 TRIB—Telephony Routing Information Base
 TRIP—Telephony Routing over IP
 TSAP—Transport Service Access Point
 TSAPI —Telephony Server API
 TTL—Time to Live

U

UA—User Agent
 UAC—User Agent Client
 UAS—User Agent Server
 UDP—User Datagram Protocol
 UL—Underwriters Laboratories
 UMTS—Universal Mobile Telecommunications Systems
 UNI—User-to-Network Interface
 UPS—Uninterruptible Power Supply
 UPT—Universal Portable Telephone
 URI—User Resource Identifier
 URL—Uniform Resource Location
 UTC—Coordinated Universal Time
 UTP—Unshielded Twisted Pair

V

VAC—Volts Alternating Current
 VANS—Value Added Network Services
 VAR—Value Added Reseller

VarBind—Variable Binding
VBR—Variable Bit Rate
VC—Virtual Channel (ATM)/Virtual Container (SDH)
VCC—Virtual Channel Connection (ATM)
VCI—Virtual Channel Identifier
VDC—Volts Direct Current
VFD—Vacuum Florescent Display
VLAN—Virtual Local Area Network
VLL—Virtual Leased Lines
VoIP—Voice Over Internet Protocol
VP—Virtual Path
VPC—Virtual Path Connection
VPI—Virtual Path Identifier
VPN—Virtual Private Network
VSA—Vendor-specific Attribute (RADIUS extension)
VTOA—Voice and Telephony over ATM

W

WAN—Wide Area Network
WLL—Wavelength Division Multiplex

X

XE—Translation Engine
XML—Extensible Markup Language
XSM—External Search Machine

Y

(None to list.)

Z

(None to list.)

Appendix A: RTC Support

This appendix summarizes real-time configuration (RTC) support status for the Net-Net SBC. The table below lists which configuration elements are supported by RTC and which are not.

ACLI Supported Configuration Elements	ACLI Unsupported Parameters
Access Control	
Accounting Config	
Authentication	
Certificate Record	
Class Profile	
Codec Policy	
DNS ALG Service	
DNS Config	
Enum	
External Policy Server	
H.323	The following H.323 stack subelement parameters are not RTC supported in that, if you save and activate a configuration, calls already in progress will be dropped. A reboot is required. • state • isgateway • realm-id • assoc-stack • options • proxy-mode • local-ip • max-calls • max-channels • registration-ttl • terminal-alias • prefixes • ras-port • q931-port • auto-gk-discovery • multicast • gatekeeper • h245-tunneling • gk-identifier • alternate-transport • q931-max-calls • filename
Host Route	

ACLI Supported Configuration Elements	ACLI Unsupported Parameters
IPSEC	
IWF	
Licensing	
Local Policy	
Local Response Map	
Local Routing Config	
Media Manager	
Media Policy	
Media Profile	
MGCP	
Network Interface	
Net Management Control	
Network Parameters	
NTP Sync	
Packet Trace Config	
Q850 SIP Map	
Realm Config	
Session Agent	
Session Group	
Session Router	
Session Translation	
Session Constraints	
SIP Config	
SIP Feature	
SIP Interface	
SIP Manipulation	
SIP NAT	
SIP Q850 Map	
SIP Response Map	
SNMP	
Static Flow	
Steering Pool	
Surrogate Agent	
System	

ACLI Supported Configuration Elements	ACLI Unsupported Parameters
TLS Global	
TLS Profile	
Translation Rules	
Trap Receiver	

