



Net-Net® 4000 ACLI Configuration Guide

Release Version S-C6.2.0

Acme Packet, Inc.
71 Third Avenue
Burlington, MA 01803 USA
t 781-328-4400
f 781-425-5077
www.acmepacket.com

Notices

©2002—2009 Acme Packet, Inc., Burlington, Massachusetts. All rights reserved. Acme Packet®, Session Aware Networking®, Net-Net®, and related marks are registered trademarks of Acme Packet, Inc. All other brand names are trademarks, registered trademarks, or service marks of their respective companies or organizations.

Patents Pending, Acme Packet, Inc.

The Acme Packet Documentation Set and the Net-Net systems described therein are the property of Acme Packet, Inc. This documentation is provided for informational use only, and the information contained within the documentation is subject to change without notice.

Acme Packet, Inc. shall not be liable for any loss of profits, loss of use, loss of data, interruption of business, nor for indirect, special, incidental, consequential, or exemplary damages of any kind, arising in any way in connection with the Acme Packet software or hardware, third party software or hardware, or the documentation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above exclusions may not apply. These limitations are independent from all other provisions and shall apply notwithstanding the failure of any remedy provided herein.

Copying or reproducing the information contained within this documentation without the express written permission of Acme Packet, Inc., 71 Third Avenue, Burlington, MA 01803, USA is prohibited. No part may be reproduced or retransmitted.

Acme Packet Net-Net products are protected by one or more of the following patents: United States: 7072303, 7028092, 7002973, 7133923, 7031311, 7142532, 7151781. France: 1342348, 1289225, 1280297, 1341345, 1347621. Germany: 1342348, 1289225, 1280297, 1341345, 1347621. United Kingdom: 1342348, 1289225, 1280297, 1341345, 1347621. Other patents are pending.

About this Guide

Overview

The *Net-Net 4000 ACLI Configuration Guide* provides information about:

- Basic concepts that apply to the key features and abilities of your Net-Net SBC
- 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 SBC

About Net-Net 4000 Software Releases

Release version S-C6.2.0 is supported on the Net-Net 4000 series platforms and Net-Net 3000 series platforms. The 4000 series contains two systems: the Net-Net 4250 and the Net-Net 4500. The Net-Net 3000 series contains the Net-Net 3800. When S-C6.2.0 is compiled to run on the Net-Net 4250 system, Acme Packet calls the image S-C6.2.0. When S-C6.1.0 is compiled to run on the Net-Net 3800 or 4500 system, Acme Packet calls the image S-CX6.2.0.

Note that some features for S-C6.2.0 are supported on the Net-Net 3800 and 4500 only—meaning they are not supported for the Net-Net 4250.

Who is Acme Packet?

Acme Packet enables service providers to deliver trusted, first class interactive communications—voice, video and multimedia sessions—across IP network borders. Our Net-Net family of session border controllers satisfy critical security, service assurance and regulatory requirements in wireline, cable and wireless networks. Our deployments support multiple applications—from VoIP trunking to hosted enterprise and residential services; multiple protocols—SIP, H.323, MGCP/NCS and H.248; and multiple border points—interconnect, access network and data center.

Established in August, 2000 by networking industry veterans, Acme Packet is a public company traded on the NASDAQ and headquartered in Burlington, MA.

Related Documentation

The following table lists related documents.

Document Name	Document Description
Net-Net 4250 Hardware Installation Guide (400-0003-00)	Contains information about the components and installation of the Net-Net 4250 SBC.
Net-Net 4500 System Hardware Installation Guide (400-0101-00)	Contains information about the components and installation of the Net-Net 4500 system.
Net-Net 3800 Hardware Installation Guide (400-0118-00)	Contains information about the components and installation of the Net-Net 4500 system.
Net-Net 4000 Storage Module Installation Guide (400-0133-00)	Contains information about how to install the storage expansion module in your Net-Net 3800 or 4500.
Net-Net 4000 Configuration Guide (400-0061-00)	Contains information about the administration and software configuration of the Net-Net SBC.
Net-Net 4000 ACLI Reference Guide (400-0062-00)	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Net-Net 4000 Maintenance and Troubleshooting Guide (400-0063-00)	Contains information about Net-Net SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
Net-Net 4000 MIB Reference Guide (400-0010-00)	Contains information about Management Information Base (MIBs), Acme Packet's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Net-Net 4000 Accounting Guide (400-0015-00)	Contains information about the Net-Net SBC's accounting support, including details about RADIUS accounting.
Net-Net 4000 Administrative Security Essentials (400-0132-00)	Contains information about the Net-Net SBC's support for its Administrative Security license.
Net-Net 4000 Border Gateway Essentials (400-0087-00)	Contains the following information about the Net-Net Border Gateway (BG): conceptual, configuration, and monitoring

Technical Assistance

If you need technical assistance with Acme Packet products, you can obtain it on-line by going to support.acmepacket.com. With your customer identification number and password, you can access Acme Packet's on-line resources 24 hours a day. If you do not have the information required to access the site, send an email to tac@acmepacket.com requesting a login.

In the event that you are experiencing a critical service outage and require live assistance, contact the Acme Packet Technical Assistance Center emergency hotline:

- From the United States, Canada, and Mexico call: 1 866 226 3758
- From all other locations, call: +1 781 756 6920

Please note that a valid support/service contract with Acme Packet is required to obtain technical assistance.

Customer Questions, Comments, or Suggestions

Acme Packet is committed to providing our customers with reliable documentation. If you have any questions, comments, or suggestions regarding our documentation, please contact your Acme Packet customer support representative directly or email support@acmepacket.com.

Contact Us

Acme Packet
71 Third Avenue
Burlington, MA 01803 USA
t 781 328 4400
f 781 425 5077
www.acmepacket.com

Contents

About this Guide	iii
Overview	iii
About Net-Net 4000 Software Releases	iii
Who is Acme Packet?.....	iii
Related Documentation	iv
Technical Assistance	v
Customer Questions, Comments, or Suggestions.....	v
Contact Us.....	v
1 Net-Net SBC Basics	57
Introduction.....	57
Realms	57
What Is a Realm?	57
Nested Realms	57
Session Agents and Session Agent Groups	58
What Is a Session Agent?.....	58
SIP session agents.....	58
H.323 session agents.....	58
Why You Need Session Agents.....	58
How to Use Session Agents.....	58
What is a Session Agent Group?.....	58
High Availability (HA)	59
2 Getting Started	61
Introduction.....	61
Installation and Start-Up	61
Hardware Installation Summary	61
Connecting to Your Net-Net SBC.....	62
Local Connections and Time-outs	62
Telnet Remote Connections and Time-outs.....	62
SSH Remote Connections	63

System Boot.....	.64
Net-Net 4000 SBC Boot Parameters65
Your Net-Net 4250 and 4500 Boot Parameters65
Sample Net-Net 4250 Boot Parameters65
Sample Net-Net 4500 Boot Parameters66
Boot Parameter Definitions66
Changing Boot Parameters68
Setting Up System Basics70
New System Prompt.....	.70
NTP Synchronization70
Your Net-Net 4000 SBC Image72
Obtaining a New Image.....	.72
Using FTP to Copy an Image on Your Net-Net 4000 SBC72
System Image Filename73
Booting an Image on Your Net-Net 4000 SBC74
Booting from Flash Memory74
Booting from an External Device.....	.74
Software Licensing.....	.75
Unlicensed Net-Net 4000 SBCs.....	.77
Obtaining a License77
Trial Licenses.....	.78
ACLI Instructions and Examples for Standalone Systems78
Adding a License to a Standalone System79
Deleting a License from a Standalone System.....	.79
ACLI Instructions and Examples for HA Nodes80
Adding a License to an HA Node80
Deleting a License from an HA Node.....	.83
Expiration.....	.87
Viewing Licenses87
Licensing Information for the Net-Net 380087
Session Capacity and Your Net-Net 380088
Granularity and Oversubscription Limits.....	.88
SNMP Support for Global Registration Capacity.....	.89
Denial of Service Feature Group89
Software TLS Feature Group90
RADIUS Authentication91
How It Works91
PAP Handshake92
PAP Client Request Example.....	.92
PAP RADIUS Response93
CHAP Handshake93

CHAP Client Request Example93
CHAP RADIUS Response93
MS-CHAP-v2 Handshake93
MS-CHAP-v2 Client Request Example94
MS-CHAP-v2 RADIUS Response94
Management Protocol Behavior94
ACLI Instructions and Examples.....	.95
Global Authentication Settings95
RADIUS Server Settings.....	.96
Customizing Your ACLI Settings97
Disabling the Second Login Prompt.....	.97
ACLI Instructions and Examples.....	.97
Persistent ACLI “more” Parameter.....	.98
ACLI Instructions and Examples.....	.98
Customized Login Banner98
3 System Configuration99
Introduction.....	.99
General System Information99
System Identification99
Connection Timeouts99
Configuring General System Information100
ACLI Instructions and Examples.....	.100
System Identification100
Configuring Connection and Debug Logging Timeouts.....	.101
Physical Interfaces: Net-Net 4250 SBC.....	.101
Overview101
Types of Physical Interfaces.....	.101
Front Interfaces102
Rear Interfaces.....	.102
Before You Configure103
ACLI Instructions and Examples.....	.103
Identity and State104
Operation Type and Location104
Auto-negotiation for 10/100 Front Interfaces.....	.104
HA Configuration105
Phy Link Redundancy105
How It Works105
Caveats.....	.106
ACLI Instructions and Examples106
Physical Interfaces: Net-Net 4500 SBC.....	.107

Network Media Interfaces107
Network Management Interfaces108
Before You Configure109
ACLI Instructions and Examples.....	.109
Interface Utilization: Graceful Call Control, Monitoring, and Fault Management.....	.109
Calculation Overview.....	.109
Alarms110
ACLI Instructions and Examples.....	.110
Configuring Utilization Thresholds for Media Interfaces110
Configuring Graceful Call Control111
Network Interfaces.....	.111
Overview111
IP Configuration111
VLANs111
VLAN Network Layer Segmentation112
Overlapping Networks.....	.112
HIP112
Configuring Network Interfaces.....	.112
Special Considerations112
ACLI Instructions and Examples.....	.113
IP Configuration and Identification.....	.113
VLANs114
HIP Addresses.....	.115
SNMP116
Overview116
Basic SNMP Parameters.....	.116
SNMP Community116
Trap Receivers116
Configuring SNMP117
SNMP Configuration Overview117
ACLI Instructions and Examples.....	.117
System Wide Configuration for SNMP.....	.118
SNMP Community Configuration.....	.119
Trap Receiver Configuration120
Syslog and Process Logs121
Overview121
Process Log Messages121
Syslog and Process Logs Configuration.....	.122
ACLI Instructions and Examples.....	.122
Syslog Configuration122

Process Log Configuration.....	.123
Host Routes123
Overview123
Host Routes Example124
ACLI Instructions and Examples.....	.124
Holidays.....	.125
ACLI Instructions and Examples.....	.125
Enhanced Control of UDP and TCP Ports.....	.126
ACLI Instructions and Examples.....	.126
DNS Transaction Timeout128
Retransmission Logic128
Configuring DNS Transaction Timeout128
ACLI Instructions and Examples.....	.128
Persistent Protocol Tracing.....	.129
About Persistent Protocol Tracing.....	.129
About the Logs130
Process Logs130
Communication Logs130
Protocol Trace Logs.....	.130
ACLI Instructions and Examples.....	.130
Enabling Persistent Protocol Tracing130
System Access Control131
ACLI Instructions and Examples.....	.131
Adding an ACL for the Management Interface131
Notes on Deleting System ACLs.....	.132
System TCP Keepalive Settings.....	.132
ACLI Instructions and Examples133
Configurable TCP Timers.....	.134
ACLI Instructions and Examples.....	.134
Configuring TCP Connection Establishment134
Configuring TCP Data Retransmission.....	.135
Timer for Idle Connections136
Historical Data Recording (HDR)137
How It Works137
Protocol Use137
About the CSV File137
Collection Interval and Push137
Group Record Types138
ACLI Instructions and Examples.....	.143

Accessing the HDR Configuration Parameters143
Global Collection Settings143
HDR for an HA Node.....	.144
Collection Group Settings145
Push Receiver Settings146
Starting and Stopping HDR at the Command Line147
HDR Monitoring.....	.147
Packet Trace.....	.148
How It Works148
Packet Trace Scenarios.....	.149
Packet Trace for One Endpoint149
Packet Trace for Both Call Legs.....	.150
Packet Trace for a Net-Net SBC Signaling Address151
ACLI Instructions and Examples.....	.151
Configuring a Trace Server.....	.151
Starting a Packet Trace152
Stopping a Packet Trace152
RAMdrive Log Cleaner.....	.153
How It Works153
Applicable Settings153
Clean-Up Procedure.....	.154
Clean-Up Frequency.....	.154
ACLI Instructions and Examples.....	.154
Alarm Synchronization.....	.156
Caveats.....	.156
ACLI Instructions and Examples.....	.156
Accounting Configuration157
SIP over SCTP.....	.157
SCTP Concepts158
SCTP Overview and Comparisons158
How Is SCTP Different from TCP?158
How Is SCTP Different from UDP?159
ACLI Instructions and Examples.....	.159
Setting the SCTP Delivery Mode.....	.159
About Your Net-Net 3800/4500 and IPv6.....	.160
Licensing160
Globally Enabling IPv6160
Updated ACLI Help Text161
IPv6 Address Configuration.....	.161
Access Control162

Host Route.....	.162
Local Policy162
Network Interface.....	.162
Realm Configuration163
Session Agent163
SIP Configuration163
SIP Interface>SIP Ports163
Steering Pool.....	.164
System Configuration.....	.164
IPv6 Default Gateway164
Network Interfaces and IPv6164
Access Control List Support.....	.165
Data Entry165
DNS Support.....	.165
Homogeneous Realms.....	.166
Parent-Child Network Interface Mismatch166
Address Prefix-Network Interface Mismatch.....	.166
RADIUS Support for IPv6.....	.166
Supporting RADIUS VSAs.....	.166
4 Realms and Nested Realms169
 Introduction.....	.169
Overview169
About Realms and Network Interfaces.....	.170
About the SIP Home Realm.....	.170
About Realms and Other Net-Net SBC Functions170
 Configuring Realms.....	.171
Before You Configure171
ACLI Instructions and Examples.....	.171
Identity and IP Address Prefix.....	.171
Realm Interfaces172
Realm Service Profile172
QoS Measurement173
QoS Marking173
Address Translation Profiles174
DNS Servers174
DoS/ACL Configuration.....	.174
Enabling RTP-RTCP UDP Checksum Generation.....	.174
Aggregate Session Constraints Per Realm174
ACLI Instructions and Examples.....	.175

Admission Control Configuration175
Reserved Parameters175
Nested Realms176
Configuring Nested Realms177
ACLI Instructions and Examples178
Required Signaling Service Parameters178
Aggregate Session Constraints: Nested Realms179
Impact to Other Session Constraints and Emergency Calls180
ACLI Instructions and Examples180
Realm-Based Packet Marking181
About TOS/DiffServ181
ToS Byte181
DiffServ Byte181
Packet Marking for Media182
Configuring Packet Marking by Media Type182
ACLI Instructions and Examples: Packet Marking for Media182
Applying a Media Policy to a Realm183
Packet Marking for Signaling183
ACLI Instructions and Examples183
Configuring a Media Policy for Signaling Packet Marking183
Configuring the Class Profile and Class Policy184
Applying a Media Policy to a Realm184
SIP-SDP DCSP Marking/ToS Bit Manipulation184
How It Works185
ACLI Instructions and Examples185
Steering Pools186
Configuration Overview186
ACLI Instructions and Examples187
Multiple Interface Realms189
How It Works189
Steering Pool Port Allocation191
ACLI Instructions and Examples191
Creating a List of Network Interfaces for the Realm191
Creating Steering Pools for Multiple Interface Realms191
Media over TCP193
TCP Bearer Conditions193
TCP Port Selection193
SDP Offer Example197
Timers198
ACLI Instructions and Examples198

Restricted Media Latching199
About Latching199
Restricted Latching199
Symmetric Latching200
How it Works200
Relationship to Symmetric Latching200
Example 1.....	.201
Example 2.....	.201
ACLI Instructions and Examples.....	.201
Media Release Across SIP Network Interfaces202
Example202
ACLI Instructions and Examples.....	.203
Media Release Behind the Same IP Address204
Additional Media Management Options204
Configuring Media Release Behind the Same IP Address204
ACLI Instructions and Examples.....	.204
Bandwidth CAC for Media Release205
ACLI Instructions and Examples.....	.205
Media Release between Endpoints with the Same IP Address206
ACLI Instructions and Examples.....	.206
Media Release Behind the Same NAT IP Address206
ACLI Instructions and Examples.....	.207
Codec Reordering207
How It Works208
Preferred Codec Precedence208
ACLI Instructions and Examples.....	.208
Setting a Preferred Codec for a Realm209
Setting a Preferred Codec for a Session Agent.....	.209
Media Profiles Per Realm210
Call Admission Control and Policing210
ACLI Instructions and Examples.....	.211
About Wildcarding211
Multiple Media Profiles212
Use Case 1.....	.213
Use Case 2.....	.213
ACLI Instructions and Examples.....	.213
Peer-to-Peer MSRP TCP Stitching214

5 SIP Signaling Services.....	215
Introduction.....	.215
About the Net-Net SBC and SIP215
Types of SIP Devices.....	.215
Basic Service Models.....	.216
About B2BUA216
SIP B2BUA Peering.....	.216
B2BUA Hosted IP Services.....	.217
SIP B2BUA and L3/L5 NAT.....	.217
About SIP Interfaces.....	.217
SIP INVITE Message Processing.....	.218
Example218
Configuring the Net-Net SBC for SIP Signaling219
The Home Realm220
Overview220
SIP NAT Function.....	.220
Home Realm's Purpose220
ACLI Instructions and Examples.....	.221
SIP Interfaces222
Overview222
About SIP Ports.....	.222
Preferred SIP Port223
Proxy Mode223
Redirect Action223
SIP maddr Resolution.....	.223
ACLI Configuration and Examples225
Trust Mode225
About the Process225
Configurable Timers and Counters.....	.226
ACLI Instructions and Examples.....	.226
Configuring SIP Ports.....	.232
SIP: PRACK Interworking233
How It Works233
UAC-Side PRACK Interworking.....	.233
UAS-Side PRACK Interworking234
ACLI Instructions and Example.....	.235
Global SIP Timers235
Overview236
ACLI Instructions and Examples.....	.236
SIP Per-User CAC237

How It Works238
Per User CAC Modes238
Per User CAC Sessions.....	.238
Per User CAC Bandwidth.....	.238
Notes on HA Nodes239
ACLI Instructions and Examples.....	.239
SIP Per-Realm CAC.....	.240
How It Works240
ACLI Instructions and Examples.....	.240
Enabling Realm-Based CAC241
Viewing Realm-Based CAC Data241
SIP Options Tag Handling242
Overview242
Configuration Overview.....	.242
ACLI Instructions and Examples.....	.243
SIP Options245
Overview245
Global SIP Options.....	.245
SIP Interface Options251
SIP Session Agent Options252
SIP Realm Options253
ACLI Instructions and Examples.....	.253
Configuring Multiple Options253
Adding an Entry253
SIP Security254
Denial of Service Protection.....	.254
Levels of DoS Protection254
Configuration Overview.....	.255
SIP Unauthorized Endpoint Call Routing255
ACLI Instructions and Examples.....	.255
Configuring Security.....	.256
The SIP NAT Function257
Overview257
NAT Modes257
Adding a maddr Parameter to a URI.....	.258
About Headers258
Replacing Headers259
Mapping FQDNs.....	.259
SIP NAT Function Cookies259
userinfo260
host.....	.260

URL Parameter260
tel URL.....	.261
Configuration Overview.....	.261
SIP NAT Interface.....	.262
SIP NAT Function Policies.....	.263
ACLI Instructions and Examples.....	.264
SIP Realm Bridging268
About SIP NAT Bridging268
SIP NAT Bridge Configuration Scenarios.....	.269
Many to One Configuration.....	.270
One-to-One Configuration270
SIP NAT Bridge Configuration270
Creating a Virtual Home Network271
Many-to-One Configuration271
One-to-One Configuration272
Shared Session Agent.....	.272
SIP Hosted NAT Traversal (HNT)273
About SIP HNT.....	.273
Using HNT with Existing NAT Device273
Registering Endpoints.....	.274
Establishing Media Flows.....	.274
Prerequisites274
Keeping the NAT Binding Open.....	.274
Working with Multiple Domains.....	.277
HNT Configuration Overview.....	.278
SIP HNT Single Domain Example.....	.278
SIP HNT Multiple Domain Example.....	.278
ACLI Instructions and Examples.....	.279
Global SIP Configuration.....	.281
SIP Registration Local Expiration283
How It Works283
ACLI Instructions and Examples.....	.283
SIP HNT Forced Unregistration284
When to Use Forced Unregistration284
Caution for Using Forced Unregistration285
ACLI Instructions and Examples.....	.285
Adaptive HNT.....	.286
Overview286
Adaptive HNT Example286
Synchronize A-HNT Successful Timer to Standby287
ACLI Instructions and Examples.....	.287

SIP IP Address Hiding and NATing in XML288
How It Works288
Sample SIP NOTIFY with NATed XML288
ACLI Instructions and Examples.....	.289
SIP Server Redundancy289
Overview289
Configuration Overview.....	.289
ACLI Instructions and Examples.....	.290
Administratively Disabling a SIP Registrar291
How It Works291
Considerations for Implicit Service Route Use.....	.291
ACLI Instructions and Examples.....	.292
SIP Distributed Media Release293
Overview293
Endpoint Locations.....	.294
Location of the Encoded Information294
Example: Distributed Media Release.....	.294
Overview of SIP DMR Configuration295
ACLI Instructions and Examples.....	.296
Configuring the Realm Configuration.....	.297
Add-On Conferencing.....	.297
Overview297
Caveats.....	.297
Add-On Conferencing Scenario298
SIP B2BUA Functionality298
Contact Header Processing298
Target Mapping and Conferences298
Refer-To Header Processing299
ACLI Instructions and Examples.....	.299
SIP REFER Method Call Transfer300
How it Works300
Unsuccessful Transfer Scenarios300
Call Flows302
ACLI Instructions and Examples.....	.304
REFER-Initiated Call Transfer.....	.305
How it Works305
Supported Scenarios.....	.306
.....	.306
Call Flows307
REFER Source Routing.....	.310

ACLI Instructions and Examples.....	.310
SIP REFER: Re-Invite for Call Leg SDP Renegotiation.....	.311
Scenario.....	.311
SIP Roaming311
Overview311
Process Overview312
Using Private IPv4 Addresses312
Example 1: With a NAT Firewall312
Example 2: Without a NAT Firewall313
ACLI Instructions and Examples.....	.313
Embedded Header Support.....	.314
ACLI Instructions and Examples.....	.315
Static SIP Header and Parameter Manipulation315
Header Manipulation Rules.....	.315
Header Element Rules315
About SIP Header and Parameter Manipulation.....	.316
Role in Trunk Group URI Feature316
ACLI Instructions and Examples.....	.316
Creating SIP Header Manipulation Rulesets316
Configuring a Session Agent320
Configuring a SIP Interface320
Example 1: Stripping All Route Headers.....	.321
Example 2: Stripping an Existing Parameter and Adding a New One.....	.321
SIP HMR (Header Manipulation Rules)323
How It Works324
Guidelines for Header and Element Rules325
Precedence.....	.325
Duplicate Header Names.....	.325
Performing HMR on a Specific Header.....	.325
Multiple SIP HMR Sets.....	.325
MIME Support.....	.326
How It Works: Find and Replace All.....	.326
Escaped Characters.....	.327
New Reserved Word.....	.327
About the MIME Value Type328
Back Reference Syntax329
Notes on the Regular Expression Library329
SIP Message-Body Separator Normalization.....	.330
Best Practices.....	.330
About Regular Expressions331
Expression Building Using Parentheses332

ACLI Instructions and Examples.....	.333
Configuring SIP Header Manipulation Rules.....	.333
Configuring SIP Header Manipulation Element Rules334
Status-Line Manipulation and Value Matching.....	.336
Setting the Header Name.....	.336
Setting the Element Type.....	.336
Setting the Match Value337
Setting the Response Code Block338
Configuring MIME Support.....	.339
Testing Pattern Rules340
Configuring SIP HMR Sets341
Configuration Examples.....	.341
Example 1: Removing Headers341
Example 2: Manipulating the Request URI.....	.342
Example 3: Manipulating a Header.....	.344
Example 4: Storing and Using URI Parameters345
Example 5: Manipulating Display Names.....	.346
Example 6: Manipulating Element Parameters348
Example 7: Accessing Data from Multiple Headers of the Same Type350
Example 8: Using Header Rule Special Characters352
Example 9: Status-Line Manipulation.....	.354
Example 10: Use of SIP HMR Sets355
Example 11: Use of Remote and Local Port Information356
Example 12: Response/Status Processing.....	.357
Example 13: Remove a Line from SDP359
Example 14: Back Reference Syntax360
Example 15: Change and Remove Lines from SDP.....	.361
Example 16: Change and Add New Lines to the SDP.....	.362
Dialog-Matching Header Manipulation363
About Dialog-Matching Header Manipulations363
Inbound HMR Challenge.....	.364
Outbound HMR Challenge364
ACLI Instructions and Examples.....	.365
Built-In SIP Manipulations366
ACLI Instructions and Examples.....	.366
Testing SIP Manipulations366
HMR Import-Export367
Exporting367
Importing.....	.368
Displaying Imports368
Using FTP to Move Files.....	.368
Removing Files368

Unique HMR Regex Patterns and Other Changes368
Manipulation Pattern Per Remote Entity369
Reject Action369
ACLI Instructions and Examples370
About Counters370
SNMP Support371
Log Action372
Changes to Storing Pattern Rule Values372
Removal of Restrictions373
Name Restrictions for Manipulation Rules373
New Value Restrictions373
Dialog Transparency373
Overview373
Configuring Dialog Transparency374
ACLI Instructions and Examples374
Route Header Removal374
Configuring SIP Route Header Removal374
ACLI Instructions and Examples374
SIP Via Transparency376
How it Works376
ACLI Instructions and Examples376
Symmetric Latching377
ACLI Instructions and Examples377
Enabling RTCP Latching378
SIP Number Normalization378
How it Works378
Terminology379
Calls from IP Endpoints379
Calls from IP Peer Network379
ACLI Instructions and Examples380
Realm380
Session Agent380
SIP Port Mapping381
About SIP Port Mapping381
How SIP Port Mapping Works382
SIP Port Mapping Based on IP Address383
About NAT Table ACL Entries384
Using SIP Port Mapping384
Dynamic Configuration384
Registration Statistics385

Configuring SIP Port Mapping385
ACLI Instructions and Examples385
SIP Port Mapping for TCP and TLS387
ACLI Instructions and Examples387
SIP Configurable Route Recursion388
Example 1389
Example 2389
ACLI Instructions and Examples390
Configuring a Session Agent for SIP Route Recursion390
Configuring a SIP Interface for SIP Route Recursion391
SIP Event Package Interoperability391
ACLI Instructions and Examples392
SIP REGISTER Forwarding After Call-ID Change393
ACLI Instructions and Examples393
SIP Local Response Code Mapping393
ACLI Instructions and Examples394
Creating a SIP Response Code Map394
Assigning SIP Response Code Maps to Session Agents395
Assigning SIP Response Code Maps to SIP Interfaces395
Session Agent Ping Message Formatting396
ACLI Instructions and Examples396
SIP PAI Stripping397
SIP PAI Stripping Configuration398
ACLI Instructions and Examples399
SIP Statuses to Q.850 Reasons400
SIP-SIP Calls400
ACLI Instructions and Examples401
Calls Requiring IWF402
Default Mappings403
ACLI Instructions and Examples404
Trunk Group URIs406
Terminology407
Trunk Group URI Parameters407
Originating Trunk Group URI Parameters and Formats407
Terminating Trunk Group URI Parameters and Formats408
How It Works410
SIP Header and Parameter Manipulation410
Trunk Group Routing411
Trunk Group URIs and SIP Registration Caching411
ACLI Instructions and Examples411

Configuring SIP Manipulations412
Setting the Trunk Group URI Mode for Routing412
Configuring a Session Agent for Trunk Group URIs412
Configuring a Session Agent Group for Trunk Group URIs413
Setting a Trunk Group Context in a Realm414
Using this Feature with a SIP Interface414
Example 1: Adding Originating Trunk Group Parameters in IPTEL Format414
Example 2: Adding Originating Trunk Group Parameters in Custom Format415
Example 3: Removing IPTEL Trunk Group Names415
Example 4: Removing Custom Trunk Group Names416
Emergency Session Handling416
Emergency Session Handling Configuration Procedures417
ACLI Instructions and Examples417
Setting Policy Priority418
Fraud Prevention418
ACLI Configurations and Instructions418
SIP Early Media Suppression419
How it Works419
Example420
Early Media Suppression Support421
Call Signaling421
Suppression Duration421
About the Early Media Suppression Rule421
Session Agent Rule421
Rule Resolution421
Selective Early Media Suppression422
How It Works422
Configuring Early Media Suppression422
Configuring the Realm422
Configuring Session Agents423
Configuring Realm Groups425
SDP-Response Early Media Suppression426
How it Works for SIP-Based Addressing426
How it Works with SDP-Based Addressing426
Global Realms426
Additional Prefixes427
Using the SDP-Response Early Media Suppression Rule427
Example428
Configuring SDP-Response Early Media Suppression428
Configuring the SIP Interface429
Configuring a Realm430

SIP SDP Address Correlation431
ACLI Instructions and Examples.....	.431
SDP Insertion for (Re)INVITEs433
How It Works433
SDP Insertion for SIP INVITES433
SDP Insertion for SIP ReINVITEs434
ACLI Instructions and Examples.....	.435
Configuring SDP Insertion for SIP INVITEs.....	.435
Configuring SDP Insertion for SIP ReINVITEs436
Restricted Media Latching436
About Latching436
Restricted Latching.....	.437
Symmetric Latching437
How it Works437
Relationship to Symmetric Latching.....	.437
Example 1.....	.438
Example 2.....	.438
Configuring Restricted Latching438
ACLI Instructions and Examples.....	.438
Enhanced SIP Port Mapping440
Anonymous Requests.....	.440
ACLI Instructions and Examples.....	.440
Dynamic Transport Protocol Change441
ACLI Instructions and Examples.....	.441
SIP Privacy Extensions441
How it Works441
Privacy Types Supported442
user.....	.442
header442
id.....	.443
Examples443
Calls from Untrusted Source to Trusted Target443
Calls from Trusted to Untrusted443
Calls from Trusted to Trusted443
Configuring SIP Privacy Extensions443
Trust Mode443
Disabling the PPI to PAI Change444
SIP Registration Cache Limiting445
How It Works445
About Registration Cache Additions, Modifications, and Removals446

Registration Cache Alarm Threshold446
Notes on Surrogate Registration446
Monitoring Information446
ACLI Instructions and Examples.....	.446
SIP Registration Overload Protection447
How It Works447
ACLI Instructions and Examples.....	.448
SIP Request Method Throttling.....	.449
How It Works449
About Counters and Statistics450
ACLI Instructions and Examples.....	.450
Requirements450
Rate Constraints for SIP Interfaces450
Applying Session and Rate Constraints to a SIP Interface451
Configuring Rate Constraints for Session Agents452
SIP Delayed Media Update453
Delayed Media Update Disabled.....	.453
Delayed Media Update Enabled453
How It Works453
ACLI Instruction and Examples.....	.454
SIPconnect.....	.454
Modifications to Registration Caching Behavior455
Configuring SIP Connect Support455
Required Configuration455
Suggested Additional Configuration.....	.456
ACLI Instructions and Examples.....	.456
SIP Registration Event Package Support.....	.457
Updating Expiration Values458
ACLI Instructions and Examples.....	.458
Session Replication for Recording459
How It Works459
Globally Unique Call ID for Call Replication460
X-UCID Notes.....	.461
License Information461
CRS Capacity.....	.461
ACLI Instructions and Examples.....	.462
SIP Transport Selection.....	.463
ACLI Instructions and Examples.....	.463
uaCSTA NAT Support464
Overview464

How It Works465
ACLI Instructions and Examples.....	.465
SIP Packet Cable Multi-Media465
How It Works465
Details466
ACLI Instructions and Examples.....	.467
SIP Method-Transaction Statistic Enhancements468
How It Works468
ACLI Instructions and Examples.....	.468
Enabling the SIP Method Tracking Enhancements.....	.468
National Security and Emergency Preparedness for SIP.....	.469
How It Works469
Licensing469
Matching by NMC and by RPH.....	.469
Call Treatment.....	.471
Generating Egress RPH471
Media Treatment.....	.472
ACLI Instructions and Examples.....	.472
Setting Up and Applying RPH Policy472
Setting Up and Applying RPH Profile.....	.473
Enabling NSEP for an NMC Rule474
Global SIP Configuration Settings: Enabling NSEP475
Global SIP Configuration Settings: Enabling CAC and Congestion Control475
Global SIP Configuration Settings: Enabling ARPH Insertion476
Setting Up NSEP for Session Agents477
SIP TCP Connection Reuse477
How It Works477
ACLI Instructions and Examples.....	.477
SIP TCP Keepalive478
ACLI Instructions and Examples.....	.478
SIP TCP Keepalive for Session Agents478
SIP TCP Keepalive for SIP Interfaces479
SIP Enforcement Profile and Allowed Methods479
ACLI Instructions and Examples.....	.480
Setting Up and Enforcement Profile480
Applying an Enforcement Profile480
Local Policy Session Agent Matching for SIP.....	.482
How It Works482
ACLI Instructions and Examples.....	.485
About Wildcarding486

Monitoring.....	.487
ACLI Instructions and Examples.....	.487
Setting Up Subscribe Dialog Limits487
Applying an Enforcement Profile to a Realm.....	.488
STUN Server.....	.488
About STUN Messaging489
STUN Server Functions on the Net-Net SBC490
RFC 3489 Procedures490
rfc3489bis Procedures.....	.491
Monitoring.....	.491
ACLI Instructions and Examples.....	.491
SIP GRUU492
Contact Header URI Replacement493
Record-Route Addition493
GRUU URI Parameter Name.....	.493
ACLI Instructions and Examples.....	.494
SIP ISUP Features494
SIP Diversion to SIP-ISUP Interworking494
ACLI Instructions and Examples494
SIP-ISUP Format Version Interworking.....	.496
Details496
ACLI Instructions and Examples.....	.497
HMR for SIP-ISUP498
Changes and Additions to Equality Operators.....	.498
Reserved Words498
Changes to Action.....	.499
About MIME Rules500
ACLI Instructions and Examples500
About MIME ISUP Manipulation502
Net-Net SBC MIME ISUP Parameters502
Adding an ISUP Body to a SIP Message.....	.504
ACLI Instructions and Examples505
Configuration Example.....	.507
6 H.323 Signaling Services.....	509
Introduction.....	509
Peering Environment for H.323.....	.510
Overview	511
Signaling Modes of Operation.....	.511
Back-to-Back Gateway Signaling511
Back-to-Back Gatekeeper Proxy and Gateway512

Interworking Gatekeeper-Gateway513
Realm Bridging with Static and Dynamic Routing514
Before You Configure514
Configuring Global H.323 Settings.....	.514
ACLI Instructions and Examples.....	.514
Accessing Global H.323 Parameters514
Global H.323 Settings.....	.515
Configuring H.323 Interfaces.....	.516
ACLI Instructions and Examples.....	.516
Identity and State517
Realm and Interface Associations517
H.323 Signaling Interface Settings517
H. 323 System Resource Allocation.....	.518
Configuring H.323 Service Modes519
ACLI Instructions and Examples.....	.519
Configuring Gateway Only Settings519
Gatekeeper Proxy Settings.....	.520
H.323 Features.....	.521
Fast Start/Slow Start Translations521
Fast Start to Slow Start Translation521
Slow Start to Fast Start Translation521
Configuration Prerequisites for Slow Start/Fast Start Translations522
ACLI Instructions and Examples.....	.523
Configuring Fast Start/Slow Start Translations525
ACLI Instructions and Examples.....	.525
RFC 2833: DTMF Interworking526
About RFC 2833526
About H.245 UII526
About 2833 to H.245 UII Interworking526
About DTMF Transfer.....	.527
Preferred and Transparent 2833527
Preferred 2883 Support.....	.528
Transparent 2833 Support528
Basic RFC 2833 Negotiation Support529
H.323 to H.323 Negotiation530
Signal and Alpha Type Support.....	.530
H.323 Endpoints531
Translating H.245 UII to 2833 for H.323 Calls531
ACLI Instructions and Examples.....	.531
H.323 Registration Proxy533
H.235 Authentication Transparency533

Unique CSA Per Registered Gateway.....	.533
Virtual Call Signaling Address.....	.534
Virtual RAS Address534
RAS Message Proxy534
About Setting Port Ranges.....	.535
ACLI Instructions and Examples.....	.535
H.323 Registration Caching.....	.536
Caveats for Registration Caching537
Configuration Requirements537
ACLI Instructions and Examples538
Configuring the Gatekeeper Interface for Registration Caching539
ACLI Registration Caching Configuration Example540
H.245 Stage541
Dynamic H.245 Stage Support541
Dynamic H.245 Stage for Incoming Calls.....	.541
Dynamic H.245 Stage for Outgoing Calls.....	.542
ACLI Instructions and Examples543
H.323 HNT543
Caveats.....	.544
ACLI Instructions and Examples545
H.323 Party Number-E.164 Support.....	.545
Signaling Only Operation545
H.245546
H.225546
Maintenance Proxy Function.....	.547
ACLI Instructions and Examples547
Applying TCP Keepalive to the H.323 Interface548
Automatic Gatekeeper Discovery548
ACLI Instructions and Examples548
H.323 Alternate Routing549
Without Alternate Routing Enabled549
With Alternate Routing Enabled549
ACLI Examples and Instructions550
H.323 LRQ Alternate Routing.....	.551
How It Works552
Caveats.....	.552
ACLI Instructions and Examples553
H.323 CAC Release Mechanism554
ACLI Instructions and Examples554
H.323 Per-Realm CAC.....	.555
How It Works555
Caveats.....	.555
ACLI Instructions and Examples556

H.323 Bearer-Independent Setup556
H.323 BIS Disabled556
H.323 BIS Enabled557
ACLI Instructions and Examples557
TOS Marking for H.323 Signaling.....	.558
H.323 Codec Fallback.....	.558
Codec Fallback Disabled.....	.558
Codec Fallback Enabled559
ACLI Instructions and Examples560
H.323/TCS Media Sample Size Preservation561
ACLI Instructions and Examples562
H.323-TCS: H.245 Support for H.264 and G722.1563
ACLI Instructions and Examples563
International Peering with IWF and H.323 Calls.....	.565
ACLI Instructions and Examples565
Options566
Global H.323 Options.....	.566
H.323 Interface Options567
H.323 Stack Monitoring.....	.568
ACLI Instructions and Examples.....	.568
.....	.569
H.323 Automatic Features.....	.569
Alias Mapping569
Call Hold and Transfer570
Call Hold and Transfer: Basic Call570
Call Hold and Transfer: Music on Hold572
Call Hold and Transfer: Transfer573
Media Release for SS-FS Calls575
How It Works576
Dependencies577
Hold-and-Resume Procedure577
H.323 and IWF Call Forwarding577
Previous Behavior577
New Behavior578
How It Works578
H.323 Sample Call Flow579
H.323 NOTIFY Support579
Caveats579
H.323: H.239 Support for Video+Content579
Multiple Media Streams with the Same Payload580
Support for Generic Capabilities580
Support for H.239 Generic Messages581

Support for Miscellaneous Indication582
ACLI Signaling Mode Configuration Examples583
Configuration Fields and Values for B2BGW Signaling583
Back-to-Back Gatekeeper Proxy and Gateway585
Interworking Gatekeeper-Gateway587
Additional Information590
About Payload Types590
Payload Types for Standard Audio and Visual Encodings590
About RAS Message Treatment592
7 IWF Services595
 Introduction.....	.595
Access Network Application595
Networking Peering Application.....	.596
 How It Works597
SIP/H.323 Negotiation: H.323 Fast Start597
SIP to Fast Start H.323597
H.323 Fast Start to SIP598
SIP/H.323 Negotiation: H.323 Slow Start.....	.599
H.323 SIP to Slow Start599
H.323 Slow Start to SIP600
Status and Codec Mapping601
IWF Termination from H.323.....	.601
IWF Termination During H.323 RAS602
IWF RAS Registration Failure Code Mapping602
IWF Termination from SIP.....	.604
Q.850 Cause to H.323 Release Complete Reason605
Codec Mapping.....	.605
 IWF Service Enhancements606
SIP Redirect—H.323 LRQ Management606
Redirect—LRQ Management Sample 1607
Redirect—LRQ Management Sample 2607
Redirect—LRQ Management Sample 3608
SIP INFO and DTMF UII Management608
Mid-Session Media Change608
Enhanced Support for FAX Calls.....	.609
Early Media609
Display Name Mapping.....	.610
IWF Ringback Support.....	.610
Sample 1: In-band Ringback without Progress Message611
Sample 2: In-band Ringback with Progress Message612

Sample 3: In-band Ringback without Alerting Message613
Sample 4: Out-of-band Ringback without Progress Message614
Sample Flow 5: Out-of-band Ringback with Progress Message615
H.323 Endpoint-Originated Call Hold and Transfer.....	.616
Basic Call616
Hold617
Music On Hold619
Transfer620
Conference621
IWF Call Forwarding622
New Behavior623
How It Works623
H.323 Sample Call Flow624
Media Release for H.323 SS-FS Calls for IWF624
How It Works for H.323624
Hold-and-Resume Procedure625
Additional IWF Steps626
Dependencies626
Before You Configure627
H.323 Configuration627
SIP Configuration627
The Role of Local Policy628
Local Policy in an IWF Session Initiated with H.323628
Local Policy in an IWF Session Initiated with SIP629
Configuring Interworking629
ACLI Instructions and Examples629
DTMF Support630
ACLI Instructions and Examples631
Applying the Media Profile632
RFC 2833: DTMF Interworking633
About RFC 2833633
About H.245 UII633
About RFC 2833 to H.245 UII Interworking633
About DTMF Transfer634
Preferred and Transparent 2833635
Preferred 2883 Support635
Transparent 2833 Support636
Payload Type Handling637
Basic RFC 2833 Negotiation Support638
H.323 to H.323 Negotiation638
Signal and Alpha Type Support638

H.323 to SIP Calls639
SIP Endpoints639
H.323 Non-2833 interworking with SIP639
How H.323 to SIP Calls Work640
SIP INFO—RFC 2833 Conversion640
ACLI Instructions and Examples641
RFC 2833 Mode for H.323 Stacks641
RFC 2833 Payload for H.323641
Configuring the SIP Interface642
Configuring Session Agents642
Enabling Payload Type Handling643
DTMF Transparency for IWF645
ACLI Instructions and Examples645
RFC 2833 Packet Sequencing645
ACLI Instructions and Examples645
Enhanced H.245 to 2833 DTMF Interworking646
ACLI Instructions and Examples647
Setting the Minimum Signal Duration647
SIP Tel URI Support648
ACLI Instructions and Examples649
IWF Inband Tone Option649
ACLI Instructions and Examples649
RFC 3326 Support650
Default Mappings651
ACLI Instructions and Examples653
IWF Privacy: Caller Privacy on Unsecure Networks655
About the Presentation Indicator655
H.323 to SIP IWF Call655
Example 1: SETUP Sent from h323d to Remote H.323 Endpoints655
Example 2: INVITE from h323d to sipd656
SIP to H.323657
Example: INVITE from SIP End Point to sipd657
IWF Privacy: Caller Privacy on Secure Connections659
H.323 to SIP IWF659
Calls with Presentation Allowed659
H.323 to SIP659
Sample SETUP sent from h323d to Remote H323 Endpoints660
SIP to H.323660
Example 1: INVITE from sip EP to sipd661
Example: INVITE from sipd to h323d661
IWF Privacy Extensions for Asserted Identity in Untrusted Networks662

IWF Call Originating in H.323.....	.663
Sample H.323 Setup from a Remote Endpoint663
Sample SIP INVITE from the Net-Net SBC to a SIP Endpoint664
Before You Configure664
ACLI Instructions and Examples.....	.664
IWF Privacy for Business Trunking665
A Call Originating in H.323.....	.665
Sample SETUP Message from an H.323 Endpoint666
Sample INVITE from the Net-Net SBC to the SIP Endpoint667
A Call Originating in SIP667
Sample INVITE from a SIP Endpoint to the Net-Net SBC.....	.667
Sample SETUP from the Net-Net SBC to the H.323 Endpoint668
ACLI Instructions and Examples.....	.669
Trunk Group URIs670
Terminology670
Trunk Group URI Parameters670
Originating Trunk Group URI Parameters and Formats.....	.671
Terminating Trunk Group URI Parameters and Formats672
How It Works673
SIP Header and Parameter Manipulation.....	.674
Trunk Group Routing.....	.674
Trunk Group URIs and SIP Registration Caching675
ACLI Instructions and Examples.....	.675
Configuring SIP Manipulations.....	.675
Setting the Trunk Group URI Mode for Routing.....	.675
Configuring a Session Agent for Trunk Group URIs.....	.676
Configuring a Session Agent Group for Trunk Group URIs.....	.676
Setting a Trunk Group Context in a Realm677
Using this Feature with SIP Interface Registration Caching.....	.677
Example 1: Adding Originating Trunk Group Parameters in IPTEL Format.....	.678
Example 1: Adding Originating Trunk Group Parameters in Custom Format678
Example 2: Removing IPTEL Trunk Group Names679
Example 3: Removing Custom Trunk Group Names679
Configuring SIP Manipulations.....	.679
IWF COLP/COLR Support.....	.680
How It Works680
SIP to H.323 Calls680
H.323 to SIP Calls680
ACLI Instructions and Examples.....	.681
Options for Calls that Require the IWF.....	.682
Suppress SIP Reliable Response Support for IWF684

ACLI Instructions and Examples.....	.684
IWF Codec Negotiation: H.323 Slow Start to SIP.....	.685
ACLI Instructions and Examples.....	.685
IWF: H.245 Signaling Support for G.726685
How It Works: IWF.....	.685
ACLI Instructions and Examples.....	.686
Customized G.729 Support.....	.688
How It Works688
About Dynamic Payload Mapping689
ACLI Instructions and Examples.....	.689
International Peering with IWF and H.323 Calls689
ACLI Instructions and Examples.....	.690
IWF Codec Renegotiation for Audio Sessions690
Codec Request Change from the SIP Side691
Codec Request Change from the H.323 Side.....	.691
Exceptional Cases691
ACLI Instructions and Examples.....	.692
8 MGCP/NCS Signaling Services693
 Introduction.....	.693
MGCP/NCS Overview.....	.693
MGCP/NCS and Realms694
MGCP/NCS NAT Traversal694
MGCP/NCS Network Topology695
MGCP/NCS Configuration Overview695
Before You Configure696
ACLI Instructions and Examples.....	.696
MGCP/NCS Configuration697
 DNS Authentication698
DNS Authentication Configuration Overview.....	.699
ACLI Instructions and Examples.....	.699
Additional Parameters700
 MGCP/NCS Options.....	.701
Send Media Only701
Signaling the Source IPv4 Address of Endpoints for 911 Services702
Loose Authentication702
ACLI Instructions and Examples.....	.702
MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints702
MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints Configuration Overview	
704	

ACLI Instructions and Examples.....	.704
MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints Configuration..	.704
MGCP/NCS SuperNAT705
ACLI Instructions and Examples.....	.705
Endpoint Representation705
Endpoint Number Computation706
Endpoint not behind a NAT.....	.706
Endpoint behind a NAT706
Valid Divisors707
Endpoint Translation Mode.....	.707
Endpoint Translation Examples.....	.708
Endpoint Translation709
Unit Prefix709
ACLI Instructions and Examples.....	.709
Endpoint Translation Configuration710
Call Agent Redundancy710
Call Agent Redundancy Configuration Overview.....	.711
Defining a Set of CAs for Redundancy711
DNS Resolution for Call Agent Redundancy711
Call Agent Failover711
ACLI Instructions and Examples.....	.713
CA Redundancy Configuration713
Manually Defining a Set of CAs for Redundancy713
Enhanced Roaming (IP Address Carrying).....	.713
ACLI Instructions and Examples.....	.714
MGCP Sans Media.....	.715
Configuring MGCP Sans Media715
ACLI Instructions and Examples715
MGCP Congestion Control715
How It Works715
Alarm Information716
ACLI Instructions and Examples.....	.716
MGCP Restricted Latching.....	.717
ACLI Instructions and Examples.....	.717
MGCP Endpoint Aging.....	.718
How It Works718
Dynamic Reconfiguration.....	.718
Considerations for HA718
Deletion Smoothing718
ACLI Instructions and Examples.....	.719

MGCP Stateful Graceful Backoff719
How It Works720
ACLI Instructions and Examples.....	.720
MGCP Configurable CPU Sample Rate721
How It Works721
ACLI Instructions and Examples.....	.722
MGCP/NCS X-Keepalives723
MGCP AUEP Suppression.....	.723
ACLI Instructions and Examples.....	.723
MGCP Endpoint Aging Optimization724
How It Works724
ACLI Instructions and Examples.....	.724
MGCP Configurable Endpoint Removal725
How It Works726
ACLI Instructions and Examples.....	.726
MGCP Port Mapping726
How It Works727
Availability of Ports in the Pool727
About MGCP Port Mapping and ACLs728
Activating Your Configuration with MGPC Port Mapping Changes.....	.728
ACLI Instructions and Examples.....	.728
Monitoring Enhancements729
9 Application Layer Gateway Services	731
DNS ALG731
Overview731
Configuring DNS ALG Service732
Before You Configure732
ACLI Instructions and Examples.....	.733
Identity, Realm, and Interface Addresses733
DNS Server Attributes734
DNS Transaction Timeout735
ACLI Instructions and Examples.....	.735
H.248 ALG736
Sample Application736
Gateway Masquerading737
Handoff Support.....	.737
Licensing738
ACLI Instructions and Examples.....	.738
Enabling Global H.248 Functionality739

Configuring the H.248 MGC	740
Configuring the H.248 MG	741
10 Session Routing and Load Balancing.....	743
Introduction.....	743
Routing Overview.....	743
Session Agents, Session Groups, and Local Policy	743
About Session Agents	744
SIP Session Agents.....	744
Session Agent Status Based on SIP Response	745
SIP Session Agent Continuous Ping.....	746
How It Works	746
ACLI Instructions and Examples.....	746
H.323 Session Agents.....	747
Overlapping H.323 Session Agent IP Address and Port.....	748
Managing Session Agent Traffic	748
About Session Agent Groups	749
SIP Session Agent Group Recursion.....	750
About Local Policy	751
Routing Calls by Matching Digits	751
SIP and H.323 Interworking	752
Route Preference.....	752
DTMF-Style URI Routing.....	753
SIP Routing	753
Limiting Route Selection Options for SIP.....	753
About Loose Routing	753
About the Ingress Realm	754
About the Egress Realm.....	754
Ping Message Egress Realm Precedence	754
Normal Request Egress Realm Precedence	755
ACLI Instructions and Examples.....	755
About SIP Redirect	755
Proxy Redirect	755
Tunnel Redirect.....	755
SIP Method Matching and To Header Use for Local Policies.....	756
SIP Methods for Local Policies	756
Routing Using the TO Header.....	757
H.323 Routing	758
Egress Stack Selection	758
Static Stack Selection	758

Policy-Based Stack Selection758
Registration Caching759
Gatekeeper Provided Routes.....	.760
Back-to-Back Gateway.....	.760
Back-to-Back Gatekeeper and Gateway.....	.760
Interworking Gatekeeper/Gateway.....	.761
Load Balancing762
Configuring Routing763
Configuration Prerequisite.....	.763
Configuration Order.....	.763
ACLI Instructions and Examples.....	.763
Configuring Session Agents.....	.763
Configuring Session Agent Groups.....	.773
SAG Matching for LRT and ENUM775
Configuring Local Policy775
Local Policy Matching for Parent Realms.....	.779
SIP Session Agent DNS-SRV Load Balancing780
ACLI Instructions and Examples.....	.781
Answer to Seizure Ratio-Based Routing781
How It Works782
Configuring ASR Constraints782
ACLI Instructions and Examples.....	.782
ENUM Lookup784
How ENUM Works784
Translating the Telephone Number784
About NAPTR Records.....	.785
About the Net-Net SBC ENUM Functionality.....	.785
Configurable Lookup Length.....	.785
UDP Datagram Support for DNS NAPTR Responses.....	.785
Custom ENUM Service Type Support786
ENUM Failover and Query Distribution.....	.786
ENUM Query Distribution.....	.786
Failover to New enum-config786
ENUM Server Operation States786
Server Availability Monitoring.....	.787
ENUM Server IP Address and Port.....	.787
Caching ENUM Responses787
Source URI Information in ENUM Requests787
Operation Modes788
Stateless Proxy Mode788
Transaction Stateful Proxy788

Session Stateful Proxy.....	.789
B2BUA789
Example: ENUM Stateless Proxy.....	.789
ACLI Instructions and Examples.....	.790
ACLI Instructions and Examples790
Example792
Configuring the Local Policy Attribute.....	.793
Local Policy Example793
CNAM Subtype Support for ENUM Queries.....	.794
CNAM Unavailable Response.....	.795
SIP Profile Inheritance795
ACLI Configuration and Examples.....	.795
Local Route Tables.....	.796
How It Works796
ACLI Instructions and Examples.....	.797
Setting Up a Local Routing Configuration797
Applying the Local Routing Configuration.....	.797
Local Route Table Support for H.323 and IWF798
IWF Considerations798
ACLI Instructions and Examples798
Multistage Local Policy Routing799
Routing Stages799
Network Applications.....	.799
Multistage Routing Conceptual Example799
Multistage Routing Example 2.....	.800
Customizing Lookup Keys.....	.802
Multistage Routing Lookup Termination802
Global Local Policy Termination803
ACLI Configuration and Examples803
Maintenance and Troubleshooting.....	.804
Traps.....	.804
Routing-based RN and CIC804
ACLI Instructions and Examples.....	.805
Setting the Lookup Key805
Codec Policies for SIP805
How It Works806
Relationship to Media Profiles.....	.807
Manipulation Modes.....	.807
In-Realm Codec Manipulation808
ACLI Instructions and Examples.....	.808
Creating a Codec Policy808

Applying a Codec Policy to a Realm809
Applying a Codec Policy to a Session Agent809
In-Realm Codec Manipulations.....	.810
QoS Based Routing810
Management811
ACLI Instructions and Examples.....	.811
Configuring QoS Constraints811
Applying QoS Constraint to a Realm812
11 Number Translation813
 Introduction.....	.813
About Number Translation813
Number Translation Implementation813
Number Translation in SIP URIs.....	.814
Session Translation in H.323 Messages814
Number Translation Configuration Overview.....	.814
Translation Rules815
Session Translation.....	.816
Applying Session Translations.....	.816
Session Agent816
Realm816
 Configuring Number Translation.....	.817
ACLI Instructions and Examples.....	.817
Translation Rules.....	.817
Session Translation.....	.818
Number Translation Application818
 Other Translations819
SIP NAT Translations819
FQDN Mapping819
12 Admission Control and Quality of Service Reporting821
 Overview821
 About Call Admission Control821
Bandwidth-Based Admission Control821
Multi-Level Bandwidth Policy Nesting.....	.822
Session Capacity- and Rate-based Admission Control.....	.824
Constraints for Proxy Mode824
CAC, Policing, and Marking for non-Audio/non-Video Media.....	.824
Bandwidth CAC Fallback Based on ICMP Failure.....	.825

ACLI Instructions and Examples825
Bandwidth CAC for Aggregate Emergency Sessions826
ACLI Instructions and Example826
Admission Control for Session Agents827
ACLI Instructions and Examples827
Session Capacity827
Session Rates828
Configuring Realm Bandwidth830
SIP Admission Control830
H.323 Admission Control831
MGCP Nested Realms832
Session Agent Minimum Reserved Bandwidth832
How It Works833
ACLI Instructions and Examples833
Aggregate Session Constraints for SIP834
ACLI Instructions and Examples834
Configuring Session Constraints834
Configuring CAC, Policing, and Marking for non-Audio, non-Video Media837
Support for the AS Bandwidth Modifier837
ACLI Instructions and Examples: Setting the Media Type838
ACLI Instructions and Examples: Enabling AS Modifier Support and Headroom838
Shared CAC for SIP Forked Calls839
Bandwidth Sharing Scenarios839
ACLI Instructions and Examples840
Configuring a SIP Profile840
Applying a SIP Profile841
RADIUS Accounting Support841
Monitoring841
Conditional Bandwidth CAC for Media Release842
About Conditional Bandwidth CAC for Media Release842
Details and Conditions842
INVITEs/UPDATEs Initially Received By Net-Net SBC842
INVITEs/UPDATEs Received by Second Net-Net SBC843
Conditional Admission with Per-user CAC844
ACLI Instructions and Examples844
Configuring a SIP Profile844
Applying a SIP Profile845
Configuring Require Header Option Tag845
About QoS Reporting846
Overview846
QoS Statistics846

RADIUS Support847
Configuring QoS849
ACLI Instructions and Examples.....	.849
Network Management Controls849
How It Works850
Matching a Call to a Control Rule850
For IWF Calls.....	.851
Call Handling Determination851
Treatment Methods852
Priority Call Exemption from Policy Server Approval852
Enhanced Call Gapping.....	.853
About the Call Gapping Algorithm.....	.853
ACLI Instructions and Examples.....	.853
Configuring an Individual Control Rule854
Enabling Enhanced Call Gapping855
Applying a Network Management Control Rule to a Realm856
Accounting Configuration for QoS.....	.857
ACLI Instructions and Examples.....	.857
Account Configuration858
Account Server860
13 Static Flows863
 Introduction.....	.863
About Static Flows863
About Network Address Translation ALG.....	.864
NAPT864
TFTP864
 Configuring Static Flows.....	.866
Basic Static Flow Configuration Overview866
ACLI Instructions and Examples.....	.866
About the Static Flow Parameters866
Configuring Static Flow867
14 High Availability Nodes871
 High Availability Nodes871
 Overview871
Establishing Active and Standby Roles872
Health Score872
Switchovers872

Automatic Switchovers872
Manual Switchovers873
State Transitions873
State Transition Sequences874
HA Features874
Multiple Rear Interfaces874
Configuration Checkpointing875
Gateway Link Failure Detection and Polling875
Before You Configure876
HA Node Connections877
Virtual MAC Addresses879
ACLI Instructions and Examples880
Configuring HA Node Connections882
ACLI Instructions and Examples882
Rear Interfaces882
Media Interface Virtual MAC Addresses883
Configuring HA Node Parameters883
ACLI Instructions and Examples884
Synchronizing Configurations887
ACLI Instructions and Examples887
Using Configuration Checkpointing888
ACLI Instructions and Examples888
Manually Checking Configuration Synchronization890
ACLI Instructions and Examples890
Configuring Media Interface Link Detection and Gateway Polling891
ACLI Instructions and Examples891
Signaling Checkpointing893
SIP Signaling Checkpointing893
ACLI Instructions and Examples893
MGCP Configuration for HA Nodes894
MGCP Media Session Replication895
ACLI Instructions and Examples895
Media State Checkpointing896
ACLI Instructions and Examples896
HA Media Interface Keepalive897
How It Works897
Impact to Boot-Up Behavior897
ACLI Instructions and Examples897
RTC Notes898
HA898

Protocol-Specific Parameters and RTC898
15 Security	901
Introduction.....	.901
Security Overview.....	.901
Denial of Service Protection.....	.902
Levels of DoS Protection903
About the Process904
Trusted Path905
Address Resolution Protocol Flow.....	.905
Untrusted Path905
IP Fragment Packet Flow.....	.906
Fragment Packet Loss Prevention.....	.906
Static and Dynamic ACL Entry Limits906
Dynamic Deny for HNT.....	.907
Host and Media Path Protection Process907
Session Director Access Control907
Access Control for Hosts908
Media Access Control.....	.908
Host Path Traffic Management908
Traffic Promotion908
Malicious Source Blocking.....	.909
Blocking Actions909
Protecting Against Session Agent Overloads.....	.909
ARP Flood Protection Enhancements.....	.909
High-Capacity CAM.....	.909
Dynamic Demotion for NAT Devices910
Configuring DoS Security.....	.910
Configuration Overview.....	.910
Changing the Default Net-Net SBC Behavior911
Example 1: Limiting Access to a Specific Address Prefix Range.....	.911
Example 2: Classifying the Packets as Trusted.....	.911
Example 3: Installing Only Static ACLs911
Configuring Access Control Lists912
Host Access Policing.....	.914
Configuring ARP Flood Protection916
Access Control for a Realm916
Configuring Overload Protection for Session Agents918
Media Policing919
Policing Methods920
Session Media Flow Policing920

Static Flow Policing.....	.920
Configuration Notes.....	.920
Session Media Flow Policing920
Static Flow Policing.....	.920
Configuring Media Policing for RTP Flows921
ACLI Instructions and Examples.....	.921
Configuring Media Policing for RTCP Flows922
ACLI Instructions and Examples.....	.922
Configuring Media Policing for Static Flows922
ACLI Instructions and Examples.....	.922
RTP Payload Type Mapping.....	.923
ITU-T to IANA Codec Mapping923
SDP Anonymization.....	.924
ACLI Instructions and Examples.....	.924
TCP Synchronize Attack Prevention925
About SYN.....	.925
Server Vulnerability925
Configuring TCP SYN Attack Prevention.....	.925
Transport Layer Security926
The Net-Net SBC and TLS926
TLS Features927
Domestic and International Versions927
Supported Encryption927
TLSv1 Ciphers927
Mapping SSL3 to TLSv1 Ciphers.....	.928
Signaling Support.....	.928
DoS Protection928
Endpoint Authentication929
Key Usage Control930
Key Usage List.....	.930
Extended Key Usage List930
Configuring TLS.....	.931
Process Overview931
Configuring Certificates931
Configuring the Certificate Record931
Generating a Certificate Request.....	.932
Importing a Certificate Using the ACLI933
Importing a Certificate Using FTP.....	.935
Configuring a TLS Profile.....	.935
Applying a TLS Profile936
Reusing a TLS Connection.....	.937
Keeping Pinholes Open at the Endpoint937

Viewing Certificates937
Brief Version937
Detailed Version938
Denial of Service for TLS939
ACLI Instructions and Examples.....	.939
Configuration the SIP Interface.....	.939
Configuring the SIP Configuration.....	.940
Configuring the Realm941
TLS Session Caching.....	.942
ACLI Instructions and Examples.....	.942
Untrusted Connection Timeout for TCP and TLS.....	.943
Caveats.....	.943
ACLI Instructions and Examples.....	.943
Online Certificate Status Protocol.....	.944
Caveats.....	.944
ACLI Instructions and Examples.....	.944
Key Exchange Protocols947
IKEv1 Protocol947
ACLI Instructions and Examples.....	.947
IKEv1 Global Configuration.....	.948
DPD Protocol Configuration950
IKEv1 Interface Configuration951
IKEv1 Security Association Configuration952
IPsec Security Policy Configuration.....	.956
SDP Session Description Protocol957
Protocol Overview957
Licensing and Hardware Requirements959
Operational Modes.....	.959
Single-Ended SRTP Termination.....	.959
Back-to-Back SRTP Termination.....	.960
SRTP Pass-Thru960
ACLI Instructions960
SDES Profile Configuration961
Media Security Policy Configuration.....	.962
Assign the Media Security Policy to a Realm963
ACLI Example Configurations.....	.964
Single-Ended SRTP Termination Configuration964
Back-to-Back SRTP Termination Configuration965
SRTP Pass-Thru Configuration966
Security Policy.....	.968
Modified ALCI Configuration Elements.....	.969

Multimedia Internet KEYing Protocol.....	.970
Protocol Overview970
Licensing and Hardware Requirements972
Operational Modes.....	.972
Single-Ended SRTP Termination.....	.972
Back-to-Back SRTP Termination.....	.973
SRTP Pass-Thru973
ACLI Instructions973
MIKEY Profile Configuration.....	.974
Media Security Policy Configuration.....	.975
Assigning the Media Security Policy to a Realm977
ACLI Example Configurations.....	.977
Single-Ended SRTP Termination Configuration977
Back-to-Back SRTP Termination Configuration978
SRTP Pass-Thru Configuration980
Security Policy981
Modified ALCI Configuration Elements.....	.982
IPSec Support984
Supported Protocols984
AH vs. ESP.....	.984
Tunnel Mode vs. Transport Mode984
Cryptographic Algorithms984
IPSec Implementation985
Outbound Packet Processing.....	.985
Security Policy985
Fine-grained policy Selection.....	.986
Security Associations986
Secure Connection Details.....	.986
Inbound Packet Processing987
IP Header Inspection987
SA Matching988
Inbound Full Policy Lookup.....	.988
HA Considerations988
Packet Size Considerations988
IPSec Application Example989
ACLI Instructions and Examples.....	.990
Configuring an IPSec Security Policy990
Defining Outbound Fine-Grained SA Matching Criteria991
Configuring an IPSec SA992
Defining Criteria for Matching Traffic Selectors per SA992
Defining Endpoints for IPSec Tunnel Mode.....	.993
Real-Time IPSec Process Control994
Key Generation.....	.994

IDS Reporting994
IDS Licensing994
Basic Endpoint Demotion Behavior995
Endpoint Demotion Reporting995
SNMP Reporting.....	.995
HDR Reporting.....	.996
Endpoint Demotion SNMP Traps.....	.996
Endpoint Demotion Syslog Message996
ACLI Configuration and Examples.....	.997
Endpoint Demotion due to CAC coverage997
CAC Attributes used for Endpoint Demotion997
Authentication Failures used for Endpoint Demotion.....	.998
ACLI Configuration and Examples.....	.998
Maintenance and Troubleshooting.....	.998
show sipd acls998
show mgcp acls999
16 Lawful Intercept	1001
Introduction.....	1001
Recommendations	1002
Interoperability with SS8	1002
Interoperability with Verint	1003
Interoperability Using a Dynamic Trigger by CMS	1003
Interoperability Using ALIP.....	1004
Interoperability Using X1, X2, X3	1004
17 External Policy Servers	1005
Introduction.....	1005
Call Admission Control.....	1005
Implementation Features	1006
Bandwidth Negotiation	1006
COPS connection	1007
COPS Failures.....	1007
Failure Detection.....	1007
Failure Recovery	1007
COPS PS Connection Down	1007
Net-Net High Availability Support for COPS	1008
COPS Debugging.....	1008
Configuring COPS	1009

ACLI Instructions and Examples.....	1009
Connectivity Location Function	1012
CLF Behavior.....	1012
P-Access-Network-Info Header Handling.....	1013
CLF Re-registration	1013
CLF Failures	1014
CLF Emergency Call Handling	1014
HA Functionality.....	1015
CLF Debugging.....	1015
Configuring CLF.....	1015
ACLI Instructions and Examples.....	1015
Diameter: CAC/RACF.....	1017
Diameter Connection.....	1017
Diameter Heartbeat	1017
Diameter Failures	1018
Application IDs and Modes	1018
Bandwidth-Based Call Admission Control.....	1019
Implementation Features.....	1019
Bandwidth Negotiation	1020
Session Lifetime	1021
DIAMETER AAR Query Post SDP Exchange.....	1021
Net-Net High Availability Support for CAC	1021
ACLI Instructions and Examples.....	1021
Configuring a Realm for Diameter Support	1022
Configuring the External Bandwidth Manager	1022
Configuring Media Profiles for Diameter Support: CAC Scenario.....	1024
Subscriber Information AVP	1025
Subscription ID AVP	1025
Subscription-Id-Type	1025
Subscription-Id-Data	1026
ACLI Instructions and Example.....	1026
CAC Debugging	1026
Diameter: CLF.....	1027
Diameter Connection.....	1027
Rx Interface Details.....	1027
Non-Priority Call Handling.....	1027
Priority Call Handling.....	1028
Gq Interface.....	1029
The Proxy Bit	1029
Diameter Failures	1029
Diameter: Connectivity Location Function.....	1029

CLF Behavior	1030
P-Access-Network-Info Header Handling	1031
CLF Re-registration	1031
CLF Failures	1031
CLF Emergency Call Handling	1031
HA Functionality	1032
ACLI Instructions and Examples	1032
SIP Interface Configuration for CLF Support	1032
External Policy Server for Use with a CLF	1033
CLF Debugging	1034
Diameter e2	1035
How It Works: CLF	1035
CLF Experimental Result Handling	1035
CLF Result Code Handling	1036
How It Works: RACF Experimental Result Handling	1036
About Realms and e2 Enhancements	1036
Destination Realms	1036
Origination and Host Realms	1037
ACLI Instructions and Examples	1037
Setting the Destination Realm AVP Format	1037
Setting the Domain Name Suffix for Origin-Realm and Origin-Host AVPs	1037
Optional AVP Support	1038
About the Transport-Class AVP	1038
Configuring Optional AVPs	1039
Configuring Diameter STR Timeouts	1039
ACLI Instructions and Examples	1039
Diameter Destination Realm AVP	1040
ACLI Instructions and Examples	1040
18 IMS Support	1043
Net-Net SBC IMS Support	1043
Net-Net SBC Access Border Functions	1043
Net-Net SBC Interconnect Border Functions	1043
IMS Access Border Functions	1044
P-CSCF Functions	1044
A-BGF Functions	1044
Resource and Admission Control (RACS) Functions	1045
IMS Interconnect Border Functions	1046
Interworking Function (IWF)	1046
Interconnect Border Control Function (I-BCF)	1046

Interconnect-Border Gateway Function (I-BGF)	1046
IMS Path and Service Route Header Support	1046
Path Header	1046
Service Route Header.....	1047
Summary	1047
Configuring Path and Service Route Headers	1048
ACLI Instructions and Examples	1048
IMS Support for Private Header Extensions for 3GPP.....	1049
P-Associated-URI Header	1049
P-Asserted-Identity Header.....	1049
P-Asserted-Identity Header Handling	1050
Configuring P-Asserted-Identity Header for Session Agents	1050
ACLI Instructions and Examples	1050
P-Called-Party-ID Header.....	1051
IMS Charging Headers.....	1051
P-Charging-Vector	1051
P-Charging-Vector Header Example	1052
P-Charging-Function-Address	1052
P-Charging-Function-Address Header Example.....	1054
RADIUS Accounting of Charging Headers	1054
Configuring P-Charging-Vector Processing for SIP Interfaces	1054
ACLI Instructions and Examples	1054
P-Visited-Network-ID Header	1056
Configuring P-Visited-Network-ID Header Handling for SIP Interfaces.....	1056
ACLI Instructions and Examples	1056
Surrogate Registration	1057
Integrating with IMS	1057
How it Works	1058
Registration	1059
Routing Calls from the IMS Core	1059
SIP	1059
H.323	1060
Routing Calls from the IP-PBX	1060
Configuring Surrogate Registration	1060
Example	1062
SIP Surrogate Registration Enhancements.....	1062
Without Enhancements	1062
With Enhancements	1063
Configuring the Retry Mechanism	1063
Configuring the Count Start	1064
IMS Implicit Service Route.....	1064

How It Works	1065
ACLI Instructions and Examples.....	1065
Notes About Upgrading.....	1066
IMS Charging Vector Mode Adaptation	1067
ACLI Instructions and Examples.....	1067
IMS: P-CSCF Endpoint Identification Using Address and Port	1067
ACLI Instructions and Examples.....	1068
IMS-AKA	1068
Requirements	1068
Monitoring.....	1068
ACLI Instructions and Examples.....	1069
Setting Up an IMS-AKA Profile	1069
Setting Up an IPSec Profile for IMS-AKA Use.....	1070
Enabling IMS-AKA Support for a SIP Interface.....	1070
Applying an IMS-AKA Profile to a SIP Port.....	1071
SIP, IMS, P-CSCF: P-Asserted Identity in Responses.....	1071
Important Notes	1072
ACLI Instructions and Examples.....	1072
E-CSCF Support.....	1072
Service URN Support	1072
E-CSCF Configuration Architecture	1072
CLF Connectivity	1073
NMC Emergency Call Control.....	1073
Local Policy	1073
Emergency LRT.....	1073
CLF Response Failure.....	1074
ACLI Instructions and Examples.....	1074
Maintenance and Troubleshooting.....	1075
Acronym List.....	1077
General Use Acronyms.....	1077
A	1077
B	1078
C	1078
D	1080
E	1081
F	1081
G	1081
H	1082
I	1082

J.....	1083
K.....	1083
L.....	1083
M	1084
N.....	1085
O.....	1085
P	1086
Q.....	1087
R.....	1087
S.....	1088
T	1089
U.....	1090
V.....	1090
W	1091
X.....	1091
Y.....	1091
Z.....	1091
Signaling Protocol Acronyms.....	1091
H.323	1091
MGCP	1092
SIP	1092

Introduction

This chapter introduces some basic concepts that apply to the key features and abilities of your Net-Net SBC. It is necessary that you understand the information included in this chapter to comprehend the ways to configure your Net-Net SBC. This chapter only provides a high level overview of some important Net-Net SBC concepts. Please refer to each chapter for complete descriptions of these concepts and the procedures for their configuration.

Realms

What Is a Realm?

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

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

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

Nested Realms

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

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

Session Agents and Session Agent Groups

What Is a Session Agent?

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

SIP session agents

SIP session agents can include the following:

- Softswitches
- SIP proxies
- Application servers
- SIP gateways

H.323 session agents

H.323 session agents can include the following:

- gatekeepers
- gateways
- MCUs

Why You Need Session Agents

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

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

How to Use Session Agents

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

What is a Session Agent Group?

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

Session agent groups can be logically equivalent to the following:

- Application server cluster
- Media gateway cluster
- Softswitch redundant pair

- SIP proxy redundant pair
- Gatekeeper redundant pair

High Availability (HA)

Net-Net SBCs are deployed in pairs to deliver continuous high availability (HA) for interactive communication services. The HA design guarantees that no stable calls are dropped in the event of any single point failure. Furthermore, the Net-Net SBC HA design provides for full media and call state to be shared across an HA node. The solution uses a VRRP-like design, where the two systems share a virtual MAC address and virtual IPv4 address for seamless switchovers.

In the HA pair, one Net-Net SBC is the primary system, and is used to process signaling and media traffic. The backup system remains fully synchronized with the primary system's session status. The primary system continuously monitors itself for connectivity and internal process health. If it detects service-disrupting conditions or degraded service levels, it will alert the backup Net-Net SBC to become the active system.

Introduction

Prior to configuring your Net-Net 4000 SBC 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 4000 SBC using a console connection, Telnet, or SSH (secure shell)
- Become familiar with the Net-Net 4000 SBC's boot parameters and how to change them if needed
- Obtain, add, and delete Net-Net 4000 SBC software licenses
- Load and activate a Net-Net 4000 SBC software image
- Choose a configuration mechanism: ALCI, Net-Net EMS, or ACP/XML
- Enable RADIUS authentication
- Customize your login banner

Installation and Start-Up

After you have completed the hardware installation procedures outlined in the *Net-Net 4250 Hardware Installation Guide* or *Net-Net 4500 Hardware Installation Guide*, you are ready to establish a connection to your Net-Net 4000 SBC. Then you can load the Net-Net 4000 SBC software image you want to use and establish basic operating parameters.

Hardware Installation Summary

Installing your Net-Net 4000 SBC 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 4000 series hardware installation guides.

1. Unpacking the Net-Net SBC
2. Installing the Net-Net SBC into your rack
3. Installing power supplies
4. Installing fan modules
5. Installing physical interface cards
6. Cabling the Net-Net 4000 SBC

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 4000 SBC.

Connecting to Your Net-Net SBC

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

Note: By default, Telnet and FTP connections to your Net-Net SBC are enabled.

Local Connections and Time-outs

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

One end of the cable plugs into your terminal, and the other end plugs into the RJ-45 port behind the Net-Net 4000 SBC's front flip-down door.

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

1. Set the connection parameters for your terminal to the default boot settings:
 - 1a. Baud rate: 115,200 bits/second
 - 1b. Data bits: 8
 - 1c. Parity: No
 - 1d. Stop bit: 1
 - 1e. Flow control: None
2. Use a serial cable to connect your PC to the Net-Net SBC. The serial port on the Net-Net 4000 SBC is located behind the flip-down door on the front panel of the chassis.
3. Power on your Net-Net SBC.
4. Enter the appropriate password information when prompted to log into User mode of the ACLI.

You can control the amount of time it takes for your console connection to time out by setting the **console-timeout** parameter in the system configuration. If your connection times out, the login sequence appears again and prompts you for your passwords. The default for this field is 0, which means that no time-out is being enforced. For more information, refer to this guide's *System Configuration* chapter.

Telnet Remote Connections and Time-outs

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

The Net-Net SBC can support up to five concurrent Telnet sessions. However, only one user can carry out configuration tasks at one time.

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

To Telnet to your Net-Net SBC, you need to know the IPv4 address of its administrative interface (wancom 0). The wancom0 IPv4 address of your Net-Net SBC is found by checking the **inet on ethernet** value in the boot parameters or look at the front panel display.

You can manage the Telnet connections to your Net-Net SBC by setting certain ACLI parameters and by using certain commands:

- To set a time-out due to inactivity, use the **telnet-timeout** parameter in the system configuration. You can set the number of seconds that elapse before the

Telnet connection is terminated. The default for this field is 0, which means that no time-out is being enforced. For more information, refer to this guide's *System Configuration* chapter.

- To view the users who are currently logged into the system, use the ACCLI **show users** command. You can see the ID, timestamp, connection source, and privilege level for active connections.
- From Superuser mode in the ACCLI, you can terminate the connections of other users in order to free up connections. Use the **kill user** command with the corresponding connection ID.
- From Superuser mode in the ACCLI, you can globally enable and disable Telnet connections to the Net-Net SBC.
 - As mentioned above, Telnet service is enabled by default on your Net-Net SBC.
 - To disable Telnet, type the **management disable telnet** command at the Superuser prompt and reboot your system. The Net-Net SBC then refuses any attempts at Telnet connections. If you want to restart Telnet service, type **management enable telnet**.
- If you reboot your Net-Net SBC from a Telnet session, you lose IP access and therefore your connection.

SSH Remote Connections

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

The Net-Net SBC supports three concurrent SSH and/or SFTP sessions.

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

To initiate an SSH connection to the Net-Net SBC without specifying users and SSH user passwords:

1. Open your SSH client (with an open source client, etc.).
2. At the prompt in the SSH client, type the **ssh** command, a <Space>, the IPv4 address of your Net-Net SBC, and then press <Enter>.

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

You can also create connections to the Net-Net SBC using additional username and password options.

To initiate an SSH connection to the Net-Net SBC with an SSH username and password:

1. In the ACLI at the Superuser prompt, type the **ssh-password** and press <Enter>. Enter the name of the user you want to establish. Then enter a password for that user when prompted. Passwords do not appear on your screen.
- ```
ACMEPACKET# ssh-password
SSH username [saved]: MJones
Enter new password: 95X-SD
Enter new password again: 95X-SD
```
2. Configure your SSH client to connect to your Net-Net SBC's management IPv4 address using the username you just created. The standard version of this command would be:
- ```
ssh -l MJones 10.0.1.57
```
3. Enter the SSH password you set in the ACLI.
- ```
MJones@10.0.2.54 password: 95X-SD
```
4. Enter your User password to work in User mode on the Net-Net SBC. Enable Superuser mode and enter your password to work in Superuser mode.
5. A Telnet session window opens and you can enter your password to use the ACLI.

## System Boot

When your Net-Net SBC boots, the following information about the tasks and settings for the system appear in your terminal window.

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

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

## Net-Net 4000 SBC Boot Parameters

---

Boot parameters specify what information your Net-Net SBC uses at boot time when it prepares to run applications. The Net-Net SBC's boot parameters:

- Show the Net-Net SBC's IPv4 address for the management interface (wancom0)
- Allow you to set a system prompt
- Determine what software image a Net-Net 4000 SBC uses and from where it boots that image
- Sets up an external FTP server's username and password for transferring an image to the Net-Net 4000 SBC using FTP

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

Configuring boot parameters has repercussions on the Net-Net SBC's physical and network interface configurations. When you configure these interfaces, you can set values that might override the ones set for the boot parameters. If you are configuring these interfaces and you enter parameters that match ones set for the boot parameters, the Net-Net 4000 SBC warns you that your actions might change the boot parameters. If this happens when you are working with either a physical interface or a network interface configuration, the following note appears:

**NOTE:** These changed parameters will not go into effect until reboot. Also, be aware that some boot parameters may also be changed through the PHY and Network Interface Configurations.

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

Note that the samples in this chapter are primarily geared for the Net-Net 4250 SBC. Consult the [Your Net-Net 4250 and 4500 Boot Parameters \(65\)](#) section below to learn about the key differences for the boot parameters on the Net-Net 4500.

## Your Net-Net 4250 and 4500 Boot Parameters

Although the boot parameters on the Net-Net 4250 SBC and those on the Net-Net 4500 are nearly identical, there are some key differences.

- boot device—The boot device for the Net-Net 4250 should be **wancom0**. For the Net-Net 4500, it should be **eth0**.
- file name—The file name for the Net-Net 4250 normally starts with **/tffs0/**. For the Net-Net 4500, it should start with **/boot/**.

### Sample Net-Net 4250 Boot Parameters

The full set of Net-Net 4250 SBC boot parameters appears like the ones in this sample:

```
NN4250(configure)# bootparam

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

boot device : wancom0
processor number : 0
host name : acmepacket8
file name : /tffs0/nnSC600.gz
inet on ethernet (e): 10.0.1.57:fffff0000
```

```

inet on backplane (b): 0.0.0.0
host inet (h) : 10.0.1.5
gateway inet (g) : 10.0.0.1
user (u) : user
ftp password (pw) : password
flags (f) : 0x30008
target name (tn) : acmesystem
startup script (s) : 0
other (o) :

NOTE: These changed parameters will not go into effect until reboot.
Also, be aware that some boot parameters may also be changed through
the PHY and Network Interface Configurations.

NN4250(configure)#

```

## Sample Net-Net 4500 Boot Parameters

The full set of Net-Net 4500 SBC boot parameters appears like the ones in this sample:

```

NN4500(configure)# bootparam

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

boot device : eth0
processor number : 0
host name : acmepacket8
file name : /boot/nnSC600.gz
inet on ethernet (e): 10.0.1.57:fffff0000
inet on backplane (b): 0.0.0.0
host inet (h) : 10.0.1.5
gateway inet (g) : 10.0.0.1
user (u) : user
ftp password (pw) : password
flags (f) : 0x30008
target name (tn) : acmesystem
startup script (s) : 0
other (o) :

NOTE: These changed parameters will not go into effect until reboot.
Also, be aware that some boot parameters may also be changed through
the PHY and Network Interface Configurations.

NN4500(configure)#

```

## Boot Parameter Definitions

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

| Boot Parameter   | Description                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| boot device      | Management interface name and port number of the device from which an image is downloaded (e.g., wancom0 or eth0) from an external device. |
| processor number | Processor number on the backplane.                                                                                                         |
| host name        | Name of the boot host used when booting from an external device.                                                                           |

| Boot Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file name             | <p>Name of the image file to be booted; can be entered with the filename path.</p> <p>If you are booting from the flash memory, this filename must always match the filename that you designate when you FTP the image from the source to the Net-Net 4000 SBC.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                       | <p>When booting from flash memory on a Net-Net 4250, this filename must start with /tffs0/ (referring to /boot); for example, /tffs0/nnc610.gz.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                       | <p>When booting from flash memory on a Net-Net 4500, this filename must start with /boot); for example, /boot/nnc610.gz.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| inet on ethernet (e)  | <p>Internet address of the Net-Net SBC.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|                       | <p>This field can have an optional subnet mask in the form <i>inet_adrs:subnet_mask</i>. If DHCP is used to obtain the parameters, lease timing information may also be present. This information takes the form of <i>lease_duration:lease_origin</i> and is appended to the end of the field.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                       | <p>In this parameter, the subnet mask ffff0000 = 255.255.0.0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|                       | <p>When you use the ACLI <b>acquire-config</b> command, this is the IPv4 address of the Net-Net SBC from which you will copy a configuration.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| inet on backplane (b) | <p>Internet address of the backplane interface.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                       | <p>This parameter can have an optional subnet mask and/or lease timing information, such as e (inet on ethernet) does.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| host inet (h)         | <p>Internet address of the boot host used when booting from an external device.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                       | <p>Internet address of the gateway to the boot host.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|                       | <p>Leave this parameter blank if the host is on the same network.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| user (u)              | <p>FTP username on the boot host.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ftp password (pw)     | <p>FTP password for the FTP user on the boot host.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| flags (f)             | <p>Codes that signal the Net-Net SBC from where to boot. Also signals the Net-Net SBC about which file to use in the booting process. This sequence always starts with 0x (these flags are hexadecimal). The most common codes are:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                       | <ul style="list-style-type: none"> <li>• 0x08: Means that the system looks at the filename defined in the boot configuration parameters to determine where to boot from and what file to use. If the file name parameter contains /tffsX/filename, then the system boots off the flash memory (see options below). If the file name parameter just contains a filename, then the Net-Net SBC boots off the external host defined and looks for the filename in the /tftpboot directory on that host.</li> <li>• 0x10008: Same as 0x08, plus it mounts to usr/acme on the boot host defined in the boot parameters. Mounting externally to usr/acme would be useful for copying data off the Net-Net SBC to the external host over NFS.</li> <li>• 0x30008: Does all of the above, plus it makes /usr/acme on the boot host the correct directory for logs rather than locally on the Net-Net SBC.</li> <li>• 0x70008: Does all of the above, plus it stores the configuration in usr/acme on the boot host rather than in /code in the flash memory file system.</li> <li>• 0x80008: Used for source routing.</li> </ul> <p>If your requirements differ from what these flags allow, contact your Acme Packet customer support representative for further codes.</p> |

| Boot Parameter     | Description                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| target name (tn)   | Name of the Net-Net SBC as it appears in the system prompt. For example, ACMEPACKET> or ACMEPACKET#. You need to know the target name if you are setting up an HA node.<br>This name is required to be unique among Net-Net SBCs in your network. |
| startup script (s) | For Acme Packet use only.                                                                                                                                                                                                                         |
| other (o)          | For Acme Packet use only.                                                                                                                                                                                                                         |

## Changing Boot Parameters

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

**Note:** Changes to boot parameters do not go into effect until you reboot the Net-Net SBC. The samples in this section are for the Net-Net 4250; refer to the [Your Net-Net 4250 and 4500 Boot Parameters \(65\)](#) section about to learn about Net-Net 4500 boot parameters.

We strongly recommend that you use management port 0 (wancom0) as the boot interface, and that your management network be either: (a) directly a part of your LAN for management port 0 or (b) accessible through management port 0. Otherwise, your management messages may use an incorrect source address.

### To access and change boot parameters from the ACLI:

1. In Superuser mode, type configure terminal and press <Enter>. For example:  
**ACMEPACKET# configure terminal**
2. Type bootparam and press <Enter>. The boot device parameters appear.

```
ACMEPACKET(configure)# bootparam
'.' = clear field; '-' = go to previous field; ^D = quit
boot device : wancom0
```

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

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

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

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

boot device : wancom0
processor number : 0
host name : goose
file name : /tffs0/nnc600.gz /tffs0/nnc610.gz
```

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

```
ACMEPACKET(configure) #
```

4. Exit the configure terminal branch.
5. Reboot your Net-Net SBC for the changes to take effect.

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

```
ACMEPACKET# reboot force
```

The Net-Net SBC completes the full booting sequence. If necessary, you can stop the auto-boot at countdown to fix any boot parameters.

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

6. If you have configured the boot parameters incorrectly, the Net-Net SBC goes into a booting loop and an error message appears.

```
Error loading file: errno = 0x226.
```

```
Can't load boot file!!
```

If this happens, hit the space bar on your keyboard to stop the loop, find and correct your error, and reboot your system.

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

1. When the Net-Net SBC is in the process of booting, you can press the space bar on your keyboard to interrupt when you see the following message appear:

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

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

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

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

```
boot device : wancom0
```

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

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

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

```
ACMEPACKET(configure) # bootparam
```

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

```
boot device : wancom0
```

```
processor number : 0
```

```
host name : goose
```

```
file name : /tffs0/nnc510.gz /tffs0/nnc600.gz
```

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

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

The Net-Net 4000 SBC completes the full booting sequence unless you have made an error setting the boot parameters.

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

5. If you have configured the boot parameters incorrectly, the Net-Net SBC goes into a booting loop and an error message appears.

```
Error loading file: errno = 0x226.
```

```
Can't load boot file!!
```

If this happens, hit the space bar on your keyboard to stop the loop, find and correct your error, and reboot your system.

## Setting Up System Basics

---

Before configuring and deploying your Net-Net 4000 SBC, you might want to establish some basic attributes such as a system prompt, new User and Superuser passwords, and NTP synchronization.

### New System Prompt

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

### NTP Synchronization

If you are using your Net-Net 4000 SBC for time-critical processing, you might want to use NTP for time synchronization. Setting NTP synchronizes both hardware and software clocks with the NTP server you specify.

To support NTP RTC, the Net-Net 4000 SBC will reset the system clock forcefully if its skew is too far off the remote clock. This is because the system will continue to run its NTP process (to accept configuration changes) even when NTP synchronization at the time of system boot fails.

To account for instances when the difference between the system's internal clock and the remote time source is greater than the 1000-second threshold, the Net-Net 4000 automatically resets its clock to the remote time source and re-attempts synchronization. In prior releases, the Net-Net 4000 SBC simply does not use NTP until the clock is manually reset to a time closer to the remote source's and the system is rebooted.

You can only set NTP synchronization from the ACLI, although you can view it from the Net-Net EMS.

Note that you can display NTP server data using the ACLI **show ntp servers** command, and display the NTP server status using the **show ntp status**. For more information, refer to the *Net-Net 4000 Maintenance and Troubleshooting Guide*.

#### To set NTP synchronization:

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

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

```

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

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

```
ACMEPACKET(ntp-confi g)# add-server 18.26.4.105
```

3. To delete an NTP server, type **del ete-server**, a <Space>, and IPv4 address of the server you want to delete, and then press <Enter>.

```
ACMEPACKET(ntp-confi g)# del -server 18.26.4.105
```

## Your Net-Net 4000 SBC Image

---

Your Net-Net 4000 SBC 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 4000 SBC as specified in the *Net-Net Hardware Installation Guide*, establish a connection to the Net-Net 4000 SBC, and then begin to configure it. On boot up, your system displays information about certain configurations not being present. You can dismiss these displays and begin configuring your Net-Net 4000 SBC.

If you want to use an image other than the one installed on your Net-Net 4000 SBC 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 4000 SBC from various sources. You can take any one of the following actions:

- 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 SBC.

### Using FTP to Copy an Image on Your Net-Net 4000 SBC

In addition to using FTP to copy an image to your Net-Net 4000 SBC, you can also use SFTP.

The Net-Net 4000 SBC's /boot directory has 32mb available, and operating system files about approximately 9mb each. It is a best practice, therefore, to no more than two images at a time stored in this location. One of these should be the latest version.

#### To copy an image on your Net-Net 4000 SBC using FTP:

1. Go to the directory where the image is located.
  2. Check the IP address of the Net-Net 4000 SBC's management port (wancom0). (You might think of this as a management address since it is used in the management of your Net-Net 4000 SBC.)
  3. Create the connection to your Net-Net 4000 SBC. In your terminal window, type **ftp** and the IPv4 address of your Net-Net 4000 SBC's management port (wancom0), 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 /boot directory is used for the on-board system flash memory. If you do not put the image in this directory, the Net-Net 4000 SBC will not find it.
  6. From the FTP prompt:
    - 6a. Change the directory to /boot.
- ```
ftp> cd "/boot"
```

- 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, enter the **put** command, a <Space>, the name of the image file, and then press <Enter>.

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

Confirmation that the connection is opening and that transfer is taking place appears.

- 6d. After the file transfer is complete, you can quit.

```
ftp> quit
```

7. Now you are ready to boot the Net-Net 4000 SBC using the image you just transferred.

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

Alternatively, from the console you can reboot to access the boot prompt and then configure boot parameters from there.

8. In the ACLI, execute the **save-config** command in order to save your changes.
9. Reboot your Net-Net 4000 SBC.
10. Your Net-Net 4000 SBC runs through its loading processes and return you to the ACLI login prompt.

System Image Filename

The system image filename is a name you set for the image. This is also the filename the boot parameters uses when booting your system. This filename must match the filename specified in the boot parameters. When you use it in the boot parameters, it should always start with /tffs0/ to signify that the Net-Net 4000 SBC is booting from the /boot directory.

If the filename set in the boot parameters does not point to the image you want sent to the Net-Net 4000 SBC via FTP, then you could not only fail to load the appropriate image, but you could also load an image from a different directory or one that is obsolete for your purposes. This results in a boot loop condition that you can fix stopping the countdown, entering the appropriate filename, and rebooting the Net-Net 4000 SBC.

Booting an Image on Your Net-Net 4000 SBC

You can either boot your Net-Net 4000 SBC 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 go into effect, you must reboot your Net-Net 4000 SBC. Since a reboot stops all call processing, we recommend performing tasks that call for a reboot during off-peak maintenance hours. Or if your Net-Net 4000 SBCs are set up in an HA node, you can carrying out these tasks on the standby system first.

Booting from Flash Memory

Once you have installed an image, you can boot your Net-Net 4000 SBC from its flash memory. With the exception of testing an image before you install it on the flash memory, this is generally the method you use for booting.

To boot from your Net-Net 4000 SBC flash memory:

1. Confirm that the boot parameters are set up correctly, and make any necessary changes.
You can check the boot configuration parameters by accessing the **bootparam** command from the configure terminal menu.

```
ACMEPACKET# config terminal
ACMEPACKET# bootparam
```
2. Change any boot configuration parameters that you need to change. It is especially important to change the **file name** boot configuration parameter. The file name parameter needs to use the /tffs0 value so that the Net-Net 4000 SBC boots from the flash.
3. Reboot your Net-Net 4000 SBC.
4. You are be returned to the ACLI login prompt. To continue with system operations, enter the required password information.

Booting from an External Device

Booting from an external device means that your Net-Net 4000 SBC connects to a server to retrieve the boot image at boot time. Rather than using an image stored on your system's flash memory, it downloads the image from the external device each time it reboots.

When you are testing a new image before putting it on your Net-Net 4000 SBC, you might want to boot from an external device. Ordinarily, you would not want to boot an image on your Net-Net 4000 SBC this way.

To boot an image from an external device:

1. Confirm that the Net-Net 4000 SBC is cabled to the network from which you are booting. This is port 0 on the rear panel of the Net-Net 4000 SBC chassis (wancom0). The image is loaded from the source using FTP.
2. Log into the system you want to mount.
3. On the Net-Net 4000 SBC, configure the information for the boot parameters and confirm the following:

- 3a. **boot device**—device to which you will FTP
This parameter value must contain the name of the applicable management interface, and then the number of the appropriate 10/100 port. Usually, this value is `wancom0`.
 - 3b. **file name**—name on the host of the file containing the image
The image file must exist in the home directory of the “user” on the image source.
 - 3c. **host inet**—IPv4 address of the device off of which you are booting
 - 3d. **gateway inet**—IPv4 address of the gateway to use if the device from which you are booting is not on the same network as your Net-Net 4000 SBC
 - 3e. **user**—username for the FTP account on the boot host
 - 3f. **password**—password for the FTP account on the boot host
4. Reboot your Net-Net 4000 SBC.
 5. You are returned to the ACLI login prompt. To continue with system operations, enter the required password information.

Software Licensing

The components of the Net-Net 4000 SBC 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 for each of them.

Licenses can be activated and deactivated in real time, and are fully extensible and upgradable. They are tied to specific Net-Net 4000 SBCs (by serial number) and cannot be transferred from one Net-Net 4000 SBC to another. Multiple licenses can be active on the same Net-Net 4000 SBC simultaneously. If the same feature happens to be covered by more than one license, then the latest expiration date applies.

Acme Packet software licenses are aggregate. This means that once a new license is added to the original license set, the related capacity, protocol, or interface becomes part of the functionality you can configure and deploy. For example, if your original license for session capacity is 1000 and then you add a new license for 3000 sessions, your new total session capacity is 4000.

The following software components, interfaces, and features are licensed. If you do not have a license for a given component, interfaces, or feature, its configuration parameters are not visible.

License	Description
Accounting	Establishes RADIUS servers to which the Net-Net 4000 SBC can make connections and send CDRs.
ACP	Enables the Net-Net 4000 SBC to respond to ACP requests. Required for Net-Net EMS use.
Administration Security	Enables the use of Administration Security features; installation and use of this feature set should be executed with care.
External Bandwidth Management	Enables interaction with external policy servers using COPS; you need this license if you want to use the resource allocation function (RACF)

License	Description
External CLF Management	Enables interaction with external policy servers using COPS; you need this license if you want to use connectivity location function (CLF) support
External Policy Services	A combination of the External Bandwidth Management and External CFL Management licenses
H.248	Enables the H.248 ALG.
H.323	Enables H.323 signaling.
HA	Enables two Net-Net 4000 SBCs to work as an HA node so that, in case of failover, one system can take over for the other. The two systems paired as an HA node checkpoint configuration, signaling state, and media.
IDS	Enables the use of the Net-Net SBC's intrusion detection system (IDS).
IKE	Enables the use of Internet Key Exchange version 1 (IKEv1).
IPv6	Enables IPv4-IPv6 interworking on your Net-Net 3800 or 4500; pure IPv6 works on these systems without the license being present.
IPSec	Enables the use of Internet Protocol Security (IPSec).
IWF	Enables SIP<→H.323 IWF signaling. In order to run IWF between these two protocols, you must also have valid SIP and H.323 licenses.
LI	Enables lawful intercept use.
Load balancing	Establishes distribution of traffic across gateways, application servers, softswitches, etc.
MGCP	Enables MGCP/NCS signaling.
NSEP RPH	Enables support for Emergency Telecommunications Service (ETS), which gives priority treatment of National Security and Emergency Preparedness (NSEP) communications for IP network infrastructures.
QoS	Enables measurement for QoS (jitter, packet latency, and packet loss) on the Net-Net 4000 SBC.

License	Description
Routing policies	<p>Establishes routing policies on the Net-Net 4000 SBC.</p>
	<p>Release 4.1 introduces changes to the Acme Packet routing licence so that you can access more routing capability without obtaining a license. Without a routing license, you can view and set all local-policy-based parameters and specific parameters for only one policy attributes configuration (a subset of the local policy configuration). They are:</p> <ul style="list-style-type: none"> • next-hop—Next signaling host IP address • realm—Realm of next signaling • action (formerly called replace-uri)—Replace Request-URI with next hop • app-protocol—Application protocol used to signal session agent. <p>Without a routing license, the parameters noted in the list above appear in the local policy configuration instead of in the policy attributes configuration. You can also execute the ACLI test-policy command without a routing license.</p>
	<p>You need a routing license to specify multiple policy attributes per local policy, and to specify the remainder of the parameters in the policy attributes configuration, which are:</p> <ul style="list-style-type: none"> • terminate-recursion—Whether or not to terminate route recursion with this next hop. • carrier—Carrier for the policy. • start-time—Daily time this policy goes into effect. • end-time—Daily time this policy is no longer in effect. • days-of-week—Days of the week this policy is in effect. • cost—(Unitless) cost for the policy. • state—State of the local policy attributes. • media-profiles—List of media profiles to use for this policy.
Session capacity	<p>Determines the maximum number of sessions allowed by a Net-Net 4000 SBC for all protocols combined: SIP, MGCP, H.323, and SIP<→H.323 IWF (interworking). Each flow that doubles back (or hairpins) through the Net-Net 4000 SBC counts as two flows. Options for session capacity are: 250, 500, 1000, 2000, 4000, 8000, 16000, and 32000. When your Net-Net 4000 SBC reaches 100% of its capacity, an alarm is generated and a trap sent.</p>
Session Replication for Recording (SRR)	<p>Enables session replication for recording, which helps call centers record the signaling and media packets associated with their calls.</p>
SIP	<p>Enables SIP signaling.</p>

Unlicensed Net-Net 4000 SBCs

If you log into a Net-Net 4000 SBC 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 4000 SBC arrives, you will need to obtain a key to activate the licenses for functionality 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 add functionality to your Net-Net 4000 SBC, each new feature will require its own key. To obtain additional licenses for functions on your Net-Net 4000

SBC, contact your customer support or sales representative directly or at support@acmepacket.com. You can request and purchase a license for the software you want, obtain a key for it, and then activate it on your Net-Net 4000 SBC.

When you obtain licenses, you need to provide Acme Packet with the serial number of your Net-Net 4000 SBC. You can see the system's serial number by using the ACLI `show version boot` command.

Trial Licenses

We also offer trial license periods for software components, allowing you to test a feature before deploying it.

Trial licenses are available for the same components listed at the beginning of this licensing section, but they only last for preset periods. After trial licenses expire, their functionality stops and configuration selections are removed. At that time, you can either stop using that particular functionality or you can purchase a license for it.

To obtain trial licenses, contact your Acme Packet sales or customer support representative directly or at support@acmepacket.com.

ACLI Instructions and Examples for Standalone Systems

This section shows you how to add licenses and delete them from standalone Net-Net 4000 SBCs. The process for two systems making up an HA node is different, so follow the procedure relevant to your configuration; refer to the [ACLI Instructions and Examples for HA Nodes \(80\)](#) for more information.

Adding a License to a Standalone System

Once you have obtained a license key, you can add it to your Net-Net 4000 SBC and activate it.

To add and activate a license on your Net-Net 4000 SBC:

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

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

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

```
ACMEPACKET(system)# license
ACMEPACKET(license)#
```
4. Using the add command and the key generated by Acme Packet, add the license to your Net-Net 4000 SBC.

```
ACMEPACKET(license)# add sl 25o39pvtqhas4v2r2J c1oaen9e01o21b1dmh3
```
5. You can check that the license has been added by using the ACLI show command within the license configuration.

```
ACMEPACKET(license)# show
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
ACMEPACKET(license)#
```
6. To activate your license, type the activate-config command and press <Enter>. The Net-Net 4000 SBC then enables any of the processes that support associated features.

```
ACMEPACKET# activate-config
```

Deleting a License from a Standalone System

You can delete a license from your Net-Net 4000 SBC, including licenses that have not expired. If you want to delete a license that has not expired, you need to confirm the deletion.

To delete a license from the Net-Net 4000 SBC:

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

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

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

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

```

ACMEPACKET(i license)# no
feature:
1: MGCP
2: High Availability
3: Accounting
4: SIP
5: H323
6: 250 sessions, ACP
7: QOS
selection:
5. Type the number corresponding to the license you want to delete and press
<Enter>.
selection: 7
6. If the license has not expired, you are be asked to confirm the deletion.
Delete unexpired license [y/n]?: y
ACMEPACKET(i license)#
When you show the licenses, the one you deleted should no longer appear on
the list.
7. To clear the license from the system, type the activate-config command and
press <Enter>. The Net-Net 4000 SBC then disables any of the processes that
support associated features.
ACMEPACKET# activate-config

```

ACLI Instructions and Examples for HA Nodes

This section explains how to add licenses to and delete them from two Net-Net 4000 SBCs in an HA node. The most significant difference between these procedures and the ones for standalone systems is that you do not use the ACLI **activate-config** command when you change licenses for HA nodes.

Adding a License to an HA Node

To add a license to both systems in an HA node, you start with the standby system. Once you have completed the process on the standby, you make the systems failover so that the originally active system becomes the standby. Then you run the procedure on the newly standby system.

This procedure uses the designations Net-Net SBC1 (original standby) and Net-Net SBC2 (original active) to refer to the active and standby systems.

To add a license on systems in an HA node, Part 1:

1. Confirm that Net-Net SBC1 and Net-Net SBC2 are synchronized.

You must also make sure that all of the running and current configurations on Net-Net SBC1 and Net-Net SBC2 have the same number. In the examples below, all of the configuration versions are 5.

- 1a. On Net-Net SBC1 and Net-Net SBC2, use the ACLI **show health** command to make sure that all processes are synchronized.
- 1b. On Net-Net SBC1, show the current configuration version by using the ACLI **display-current-cfg-version** command. Then use the same command on Net-Net SBC2 and be sure that its current configuration version is the same as the one on Net-Net SBC1.

```

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

```

```
NETNETSBC1#
```

```
NETNETSBC2# di spl ay-current-cfg-versi on
Current configuration version is 5
NETNETSBC2#
```

- 1c. On Net-Net SBC1, show the running configuration version by using the ACLI **display-running-cfg-version** command. Then use the same command on Net-Net SBC2 and be sure that its running configuration version is the same as the one on Net-Net SBC1.

```
NETNETSBC1# di spl ay-runni ng-cfg-versi on
Running configuration version is 5
NETNETSBC1#
```

```
NETNETSBC2# di spl ay-runni ng-cfg-versi on
Running configuration version is 5
NETNETSBC2#
```

2. Now you can add a license. To begin, type **configure terminal** and press <Enter>.

```
NETNETSBC1# confi gure termi nal
NETNETSBC1(configure)#
```

3. Type **system** and press <Enter>.

```
NETNETSBC1(configure)# system
NETNETSBC1(system)#
```

4. Type **license** and press <Enter>.

```
NETNETSBC1(system)# l i cense
NETNETSBC1(l i cense)#
```

5. Using the **add** command and the key generated by Acme Packet, add the license to your Net-Net 4000 SBC.

```
NETNETSBC1(l i cense)# add sj kl 4i 45987p43hh0938hnj l ai e10983
```

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

```
NETNETSBC1(l i cense)# show
```

```
1: MGCP
2: Hi gh Avai labi lity
3: Accounti ng
4: SIP
5: H323
6: 250 sessi ons, ACP
7: QOS
NETNETSBC1(l i cense)#

```

7. Type **done** to save the added license to your configuration.

```
NETNETSBC1# done
```

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

```
NETNETSBC1(l i cense)# exi t
NETNETSBC1(system)# exi t
NETNETSBC1(configure)# exi t
NETNETSBC1#
```

9. Reboot Net-Net SBC1 by typing **reboot** and confirming you want to reboot the system.

```
NETNETSBC1# reboot
-----
WARNING: you are about to reboot this SD!
-----
```

Reboot this SD [y/n]?: **y**

10. Confirm that Net-Net SBC2 has finished rebooting and the HA node is full resynchronized using the ACLI **show health** command.

```
NETNETSBC1# show health
```

11. Trigger a switchover between the two systems in the HA node so the originally standby system assumes the active role. This means that the originally active system will go standby, and then you can add the license to that system (which completes the process).

```
NETNETSBC1# notify berpd force
```

12. Wait for Net-Net SBC2 to transition to the standby state. Confirm that it is in the standby state by using the ACLI **show health** command.

```
NETNETSBC2# show health
```

To add a license on systems in an HA node, Part 2:

1. Reconfirm that Net-Net SBC1 and Net-Net SBC2 are synchronized.

You must also make sure that all of the running and current configurations on Net-Net SBC1 and Net-Net SBC2 have the same number. In the examples below, all of the configuration versions are 5.

- 1a. On Net-Net SBC1 and Net-Net SBC2, use the ACLI **show health** command to make sure that all processes are synchronized.
- 1b. On Net-Net SBC2, show the current configuration version by using the ACLI **display-current-cfg-version** command. Then use the same command on Net-Net SBC1 and be sure that its current configuration version is the same as the one on Net-Net SBC2.

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

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

- 1c. On Net-Net SBC2, show the running configuration version by using the ACLI **display-running-cfg-version** command. Then use the same command on Net-Net SBC1 and be sure that its running configuration version is the same as the one on Net-Net SBC2.

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

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

2. Now you can add a license. To begin, type **configure terminal** and press <Enter>.

```
NETNETSBC2# configure terminal
```

- NETNETSBC2(configure)#
 3. Type **system** and press <Enter>.
 NETNETSBC2(configure)# **system**
 NETNETSBC2(system)#
 4. Type **license** and press <Enter>.
 NETNETSBC2(system)# **license**
 NETNETSBC2(license)#
 5. Using the **add** command and the key generated by Acme Packet, add the license to your Net-Net 4000 SBC.
 NETNETSBC2(license)# **add sjkl4l45987p43hh0938hnjhiale10984**
 6. You can check that the license has been added by using the ACLI **show** command within the license configuration.
 NETNETSBC2(license)# **show**
 1: MGCP
 2: High Availability
 3: Accounting
 4: SIP
 5: H323
 6: 250 sessions, ACP
 7: QOS
 NETNETSBC2(license)#
 7. Type **done** to save the added license to your configuration.
 NETNETSBC2# **done**
 8. Repeat typing **exit**, pressing <Enter> after each, until you reach the main Superuser prompt.
 NETNETSBC2(license)# **exit**
 NETNETSBC2(system)# **exit**
 NETNETSBC2(configure)# **exit**
 NETNETSBC2#
 9. Reboot Net-Net SBC1 by typing **reboot** and confirming you want to reboot the system.
 NETNETSBC2# **reboot**

 WARNING: you are about to reboot this SD!

 Reboot this SD [y/n]?: **y**

Deleting a License from an HA Node

To delete a license from both systems in an HA node, you start with the standby system. Once you have completed the process on the standby, you make the systems failover so that the originally active system becomes the standby. Then you run the procedure on the newly standby system.

This procedure uses the designations Net-Net SBC1 (original standby) and Net-Net SBC2 (original active) to refer to the active and standby systems.

To delete a license from systems in an HA node, Part 1:

1. Confirm that Net-Net SBC1 and Net-Net SBC2 are synchronized.

You must also make sure that all of the running and current configurations on Net-Net SBC1 and Net-Net SBC2 have the same number. In the examples below, all of the configuration versions are 5.

- 1a. On Net-Net SBC1 and Net-Net SBC2, use the ACLI **show health** command to make sure that all processes are synchronized.
- 1b. On Net-Net SBC1, show the current configuration version by using the ACLI **display-current-cfg-version** command. Then use the same command on Net-Net SBC2 and be sure that its current configuration version is the same as the one on Net-Net SBC1.

```
NETNETSBC1# di spl ay-current-cfg-versi on
Current configuration version is 5
NETNETSBC1#
```

```
NETNETSBC2# di spl ay-current-cfg-versi on
Current configuration version is 5
NETNETSBC2#
```

- 1c. On Net-Net SBC1, show the running configuration version by using the ACLI **display-running-cfg-version** command. Then use the same command on Net-Net SBC2 and be sure that its running configuration version is the same as the one on Net-Net SBC1.

```
NETNETSBC1# di spl ay-runni ng-cfg-versi on
Runni ng configuration version is 5
NETNETSBC1#
```

```
NETNETSBC2# di spl ay-runni ng-cfg-versi on
Runni ng configuration version is 5
NETNETSBC2#
```

2. Now you can delete a license. To begin, type **configure terminal** and press <Enter>.

```
NETNETSBC1# confi gure terminal
NETNETSBC1(configure)#
```

3. Type **system** and press <Enter>.

```
NETNETSBC1(configure)# system
```

4. Type **license** and press <Enter>.

```
NETNETSBC1(system)# li cense
NETNETSBC1(li cense)#+
```

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

```
NETNETSBC1(li cense)#+ no
feature:
1: MGCP
2: Hi gh Avai labi lity
3: Accounti ng
4: SIP
5: H323
6: 250 sessi ons, ACP
7: QOS
sel ecti on:
```

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

selection: 7

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

```
Delete unexpired license [y/n]?: y
NETNETSBC1(license)#

```

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

8. Type **done** to save your changes.

```
NETNETSBC1# done
```

9. Repeat typing **exit**, pressing <Enter> after each entry, until you reach the main Superuser prompt.

```
NETNETSBC1(license)# exit
NETNETSBC1(system)# exit
NETNETSBC1(configure)# exit
NETNETSBC1#
```

10. Reboot Net-Net SBC1 by typing **reboot** and confirming you want to reboot the system.

```
NETNETSBC1# reboot
```

```
-----  
WARNING: you are about to reboot this SD!  
-----
```

Reboot this SD [y/n]?: y

11. Trigger a switchover between the two systems in the HA node so the originally standby system assumes the active role. This means that the originally active system will go standby, and then you can delete the license to that system (which completes the process).

```
NETNETSBC1# notify berpd force
```

12. Wait for Net-Net SBC2 to transition to the standby state. Confirm that it is in the standby state by using the ACLI **show health** command.

```
NETNETSBC2# show health
```

To delete a license from systems in an HA node, Part 2:

1. Confirm that Net-Net SBC1 and Net-Net SBC2 are synchronized.

You must also make sure that all of the running and current configurations on Net-Net SBC1 and Net-Net SBC2 have the same number. In the examples below, all of the configuration versions are 5.

- 1a. On Net-Net SBC1 and Net-Net SBC2, use the ACLI **show health** command to make sure that all processes are synchronized.
- 1b. On Net-Net SBC2, show the current configuration version by using the ACLI **display-current-cfg-version** command. Then use the same command on Net-Net SBC1 and be sure that its current configuration version is the same as the one on Net-Net SBC2.

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

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

```
NETNETSBC1#
```

- 1c. On Net-Net SBC2, show the running configuration version by using the ACLI **display-running-cfg-version** command. Then use the same command on Net-Net SBC1 and be sure that its running configuration version is the same as the one on Net-Net SBC2.

```
NETNETSBC2# di spl ay-current-cfg-versi on
Current configuration version is 5
NETNETSBC2#
```

```
NETNETSBC1# di spl ay-current-cfg-versi on
Current configuration version is 5
NETNETSBC1#
```

2. Now you can delete a license. To begin, type **configure terminal** and press <Enter>.

```
NETNETSBC2# confi gure termi nal
NETNETSBC2(configure)#
```

3. Type **system** and press <Enter>.

```
NETNETSBC2(configure)# system
NETNETSBC2(system)#
```

4. Type **license** and press <Enter>.

```
NETNETSBC2(system)# l i cense
NETNETSBC2(l i cense)#+
```

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

```
NETNETSBC1(l i cense)#+ no
feature:
1: MGCP
2: Hi gh Avai labi lity
3: Accounti ng
4: SIP
5: H323
6: 250 sessi ons, ACP
7: QOS
sel ecti on:
```

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

```
sel ecti on: 7
```

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

```
Del ete unexpired l i cense [y/n]?: y
NETNETSBC2(l i cense)#+
```

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

8. Type **done** to save your changes.

```
NETNETSBC2# done
```

9. Repeat typing **exit**, pressing <Enter> after each entry, until you reach the main Superuser prompt.

```
NETNETSBC2(l i cense)#+ exit
NETNETSBC2(system)# exit
NETNETSBC2(configure)# exit
NETNETSBC2#
```

10. Reboot Net-Net SBC1 by typing **reboot** and confirming you want to reboot the system.

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

Expiration

When a license expires, you are no longer able to use the features associated with it. The Net-Net 4000 SBC automatically disables all associated processes.

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

Expired licenses appear in your Net-Net 4000 SBC ACLI displays until you delete them, though you cannot use the features associated with them. Deleting an expired license requires that you take the same steps as you do for deleting a valid one.

Viewing Licenses

There are two ways to view licenses in the ACLI.

- You can use the **show features** command at the main ACLI user prompt.
- ```
ACMEPACKET# show features
Total session capacity: 2250
Enabled protocols: SIP, MGCP, H.323, IWF
Enabled features: ACP
ACMEPACKET#
```
- Within the license menu, use the **show** command to see all licenses with detailed information.

```
ACMEPACKET(license)# show
License #1: 2000 sessions, SIP, MGCP, ACP
 no expiration
 installed at 12:34:42 APR 01 2005
License #2: H323
 expired at 23:59:59 APR 08 2005
 installed at 12:35:43 APR 01 2005
License #3: 250 sessions, IWF
 expires at 23:59:59 APR 28 2005
 installed at 12:36:44 APR 01 2005
License #4: QOS
 starts at 00:00:00 APR 08 2004
 expires at 23:59:59 OCT 27 2005
 installed at 12:37:45 APR 01 2005
Total session capacity: 2250
ACMEPACKET(license)#

```

## Licensing Information for the Net-Net 3800

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

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

For more information about Net-Net system licensing, including examples of how to install licenses, refer to the *Getting Started* chapter of the *Net-Net 4000 ACLI Configuration Guide*.

## **Session Capacity and Your Net-Net 3800**

The Net-Net 3800 supports lower session capacity than the Net-Net 4250 or 4500, with a maximum limit of 500 concurrent sessions. The following values are the session capacity values you can license for the Net-Net 3800:

- 150
- 250
- 350
- 500

Additional session capacities may be added at a later date through purchase of sessions in increments of 25, 50 or 100. Session capacity is additive in the Net-Net 3800, meaning the total number of sessions for the system is the sum of all session capacities licensed. The sum total of the licenses cannot exceed 500 sessions. The Net-Net 3800 strictly enforces this limit.

## **Granularity and Oversubscription Limits**

Only on the Net-Net 3800, the Net-Net SBC uses a 10-to-1 oversubscription limit, meaning that the system allows ten registrations for a single licensed session. The system enforces the limits across all signalling protocols.

An SNMP OID, `apSysRegistrationCapacity`, supports querying the percentage of used registration capacity. When the percentage approaches the registration capacity limit, an alarm triggers and the Net-Net 3800 sends an SNMP trap.

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

The Net-Net 3800 first checks the configurable registrations cache limits. If you have configured this value to be higher than the global oversubscription limits, the Net-Net 3800 leaves the registration cache limit value intact. However, if registrations go over the global oversubscription limit, the Net-Net 3800 will reject them, regardless of the cache limit, and the corresponding traps and alarms might not be triggered.

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

## **SNMP Support for Global Registration Capacity**

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

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

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

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

## **Denial of Service Feature Group**

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

The table below details the restrictions.

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

| Restricted Configuration Element | Restricted Parameters                                                                                                                                                        |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| realm-config                     | average-rate-limit<br>access-control-trust-level<br>invalid-signal-threshold<br>maximum-signal-threshold<br>untrusted-signal-threshold<br>nat-trust-threshold<br>deny-period |
| static-flow                      | average-rate-limit                                                                                                                                                           |

**Software TLS Feature Group**

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

## 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 SBC 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 SBC and your RADIUS server(s). In these roles:

- The Net-Net SBC restricts access and requires authentication via the RADIUS server; the Net-Net SBC 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 SBC users and authenticating them via RADIUS; the RADIUS server supports the VSA called ACME\_USER\_CLASS, which specifies what kind of user is requesting authentication and what privileges should be granted

The Net-Net SBC also supports the use of the Cisco Systems Inc.<sup>TM</sup> “Cisco-AVPair” vendor specific attribute (VSA). This attribute allows for successful administrator login to servers that do not support the Acme Packet authorization VSA. While using RADIUS-based authentication, the Net-Net SBC authorizes you to enter Superuser mode locally even when your RADIUS server does not return the ACME\_USER\_CLASS VSA or the Cisco-AVPair VSA. For this VSA, the Vendor-ID is 1 and the Vendor-Type is 9. The list below shows the values this attribute can return, and the result of each:

- shell : pri v=1 vl =15—User automatically logged in as an administrator
- shell : pri v=1 vl =1—User logged in at the “user” level, and not allowed to become an administrator
- Any other value—User rejected

## How It Works

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

- What type of authentication you want to use on the Net-Net SBC
- If you are using RADIUS authentication, you set the port from which you want the Net-Net SBC to send messages
- If you are using RADIUS authentication, you also set the protocol type you want the Net-Net SBC 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 SBC 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 SBC 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 SBC will round-robin to select a server until it finds an appropriate one; it will do the same for secondary servers.

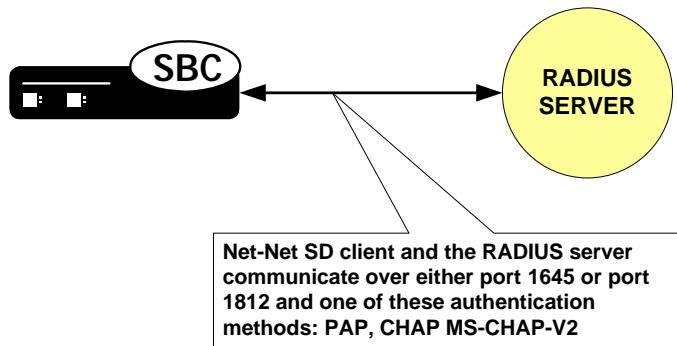
The VSA attribute assists with enforcement of access levels by containing one of the three following classes:

- 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 SBC initiates communication and proceeds with the authentication process. The authentication process between the Net-Net SBC 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 SBC.



## 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 SBC 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 SBC 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.

```
Radi us Protocol
Code: Access Request (1)
Packet identi fier: 0x5 (5)
Length: 80
Authenti cator: 0x0000024C000046B30000339F00000B78
Attribute value pairs
t:User Name(1) l:11, val ue: "TESTUSER1"
User-Name: TESTUSER1
t:Vendor Speci fic(26) l:24, vendor: Mi crosoft(311)
t: MS CHAP Challenge(11) l:18, val ue: 0000024C000046B30000339F00000B78
t:Vendor Speci fic(26) l:58, vendor: Mi crosoft(311)
t: MS CHAP2 Response(25) l:52, val ue: 00000000024C000046B30000339F00000B78...
t: NAS IP Address(4) l:6, val ue: 168. 192. 68. 8
Nas IP Address: 168. 192. 68. 8(168. 192. 68. 8)
t: NAS Port(5) l:6, val ue: 118751232
```

## MS-CHAP-v2 RADIUS Response

```
Radi us Protocol
Code: Access Accept (2)
Packet identi fier: 0x6 (6)
Length: 179
Authenti cator: 0xECB4E59515AD64A2D21FC6D5F14D0CC0
Attribute value pairs
t:Vendor Speci fic(26) l:51, vendor: Mi crosoft(311)
t: MS CHAP Success(11) l:45, val ue: 003533s33d3845443532443135453846313...
t:Vendor Speci fic(26) l:42, vendor: Mi crosoft(311)
t: MS MPPE Recv Key(17) l:36, val ue: 96C6325D22513CED178F770093F149CBBA...
t:Vendor Speci fic(26) l:42, vendor: Mi crosoft(311)
t: MS MPPE Send Key(16) l:36, val ue: 9EC9316DBFA701FF0499D36A1032678143...
t:Vendor Speci fic(26) l:12, vendor: Mi crosoft(311)
t: MS MPPE Encryption Pol i cy(7) l:6, val ue: 00000001
t:Vendor Speci fic(26) l:12, vendor: Mi crosoft(311)
t: MS MPPE Encryption Type(8) l:6, val ue: 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—Setting the user name to “user” has the same effect as using the local authentication type. For all other users, the configured RADIUS servers will be contacted. If authentication is successful, the user is granted privileges depending on the ACME\_USER\_CLASS VSA attribute.
- FTP—if you set the user name to “user” or “admin,” the user is authenticated locally. Otherwise, configured RADIUS servers are used for authentication.
- SSH in pass-through mode—When SSH is in pass through mode, the Net-Net SBC behave the same way that it does for Telnet.

- SSH in non-pass-through mode—When you create an SSH account on the Net-Net SBC, you are asked to supply a user name and password. Once local authentication succeeds, you are prompted for the ACLI user name and password. If your user ACLI name is “user,” then you are authenticated locally. Otherwise, you are authenticated using the RADIUS server. If RADIUS authentication is successful, the privileges you are granted depend on the ACME\_USER\_CLASS VSA attribute.
- SFTP in pass-through mode—if you do not configure an SSH account on the Net-Net SBC, the RADIUS server is contacted for authentication for any user that does not have the user name “user.” The Net-Net SBC uses local authentication if the user name is “user.”
- SFTP in non-pass-through mode—When you have configured an SSH account on the Net-Net SBC, user authentication takes place locally.

## ACLI Instructions and Examples

To enable RADIUS authentication and user access on your Net-Net SBC, 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# **config terminal**
2. Type **security** and press <Enter>.   
ACMEPACKET(configure)# **security**
3. Type **authentication** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(security)# **authentication**  
ACMEPACKET(authentication)#
 

From here, you can view the entire menu for the authentication configuration by typing a ?. You can set global parameters for authentication. You can also configure individual RADIUS servers; instructions for configuring RADIUS server appear in the next section.
4. **type**—Set the type of user authentication you want to use on this Net-Net SBC. The default value is **local**. The valid values are:
  - local | radius
5. **protocol**—If you are using RADIUS user authentication, set the protocol type to use with your RADIUS server(s). The default is **pap**. The valid values are:
  - pap | chap | mschapv2
6. **source-port**—Set the number of the port you want to use from message sent from the Net-Net SBC to the RADIUS server. The default value is 1812. The valid values are:
  - 1645 | 1812
7. **allow-local-authorization**—Set this parameter to **enabled** if you want the Net-Net SBC to authorize users to enter Superuser (administrative) mode locally even when your RADIUS server does not return the ACME\_USER\_CLASS VSA or the Cisco-AVPair VSA. The default for this parameter is **disabled**.

## RADIUS Server Settings

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

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

### To configure a RADIUS server to use for authentication:

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

```
ACMEPACKET(authentication)# radius-servers
ACMEPACKET(radius-servers)#
```
2. **address**—Set the remote IP address for the RADIUS server. There is no default value, and you are required to configure this address.
3. **port**—Set the port at the remote IP address for the RADIUS server. The default port is set to **1812**. The valid values are:
  - 1645 | 1812
4. **state**—Set the state of the RADIUS server. Enable this parameter to use this RADIUS server to authenticate users. The default value is **enabled**. The valid values are:
  - enabled | disabled
5. **secret**—Set the password that the RADIUS server and the Net-Net SBC 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 SBC to retry for authentication information from this RADIUS server. The default value is **3**. The valid range is:
  - Minimum—1
  - Maximum—5

If the RADIUS server does not respond within this number of tries, the Net-Net SBC marks it as dead.
8. **retry-time**—Set the amount of time (in seconds) that you want the Net-Net SBC to wait before retrying for authentication from this RADIUS server. The default value is **5**. The valid range is:
  - Minimum—5
  - Maximum—10
9. **dead-time**—Set the amount of time in seconds before the Net-Net SBC retries a RADIUS server that it has designated as dead because that server did not respond within the maximum number of retries. The default is **10**. The valid range is:
  - Minimum—10
  - Maximum—10000
10. **maximum-sessions**—Set the maximum number of outstanding sessions for this RADIUS server. The default value is **255**. The valid range is:

- Minimum—1
  - Maximum—255
11. **class**—Set the class of this RADIUS server as either primary or secondary. A connection to the primary server is tried before a connection to the secondary server is tried. The default value is **primary**. Valid values are:
- primary | secondary
- The Net-Net SBC 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 SBC 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 SBC to use with this RADIUS server. The default value is **pap**. Valid values are:
- all | pap | chap | mschapv2
- This parameter has a specific relationship to the global protocol parameter for the authentication configuration, and you should exercise care when setting it. If the authentication method that you set for the RADIUS server does not match the global authentication protocol, then the RADIUS server is not used. The Net-Net SBC 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.

## Customizing Your ACLI Settings

---

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

### Disabling the Second Login Prompt

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

### ACLI Instructions and Examples

You disable the second login prompt in the authentication configuration.

#### To disable the second login prompt:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
ACMEPACKET# **configure terminal**  
ACMEPACKET(configure)#
2. Type **security** and press <Enter>.  
ACMEPACKET(configure)# **security**  
ACMEPACKET(security)#
3. Type **authentication** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

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

Persistent ACLI “more” Parameter

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

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

ACLI Instructions and Examples

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

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


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
2. Type system and press <Enter>.
ACMEPACKET(configure)# system
ACMEPACKET(system)#
3. Type system-config and press <Enter>.
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
If you are adding this feature to an existing configuration, you need to select the configuration (using the ACLI select command) before making your changes.
```
4. **cli-more**—Set this parameter to **enabled** if you want the ACLI “more” paging feature to work persistently across console, Telnet, or SSH sessions with the Net-Net SBC. If you want to continue to set this feature on a per session basis, leave this parameter set to **disabled** (default).
5. Save and activate your configuration.

Customized Login Banner

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

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

Introduction

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

The following configurations are explained in this chapter:

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

General System Information

This section explains the parameters that encompass the general system information on a Net-Net SBC.

System Identification

Global system identification is used primarily by the Net-Net SBC to identify itself to other systems and for general identification purposes.

Connection Timeouts

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

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

After the Telnet timeout passes, the Telnet session is disconnected. You must use your Telnet program to log in to the Net-Net SBC once again to perform any further configuration activity.

After the console timeout passes, the console session is disconnected. The current session ends and you are returned to the login prompt on the console connection into the Net-Net SBC.

Configuring General System Information

This section explains how to configure the general system parameters, timeouts, and the default gateway necessary to configure your Net-Net SBC.

ACLI Instructions and Examples

To configure general system information:

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

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

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

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

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

System Identification

You must specify identification parameters for this Net-Net SBC.

Set the following parameters to configure the system identification:

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

Configuring Connection and Debug Logging Timeouts

Configure the timeouts for terminal sessions on this Net-Net SBC. These parameters are optional.

Set the following parameters to configure the connection timeouts:

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

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

The valid range is:

- Minimum—0
- Maximum—65535

Physical Interfaces: Net-Net 4250 SBC

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

Overview

The Net-Net 4250 SBC's 10/100 and GigE interfaces provide physical connections to your network. Over the front interfaces, media and signaling traffic enter and exit the Net-Net SBC. The rear interfaces are used for management and high availability (HA).

You need to configure operating parameters for physical interfaces to support them on your Net-Net SBC. These values identify the name, location, basic Ethernet properties, and HA parameters; these are all configured on a per-interface basis.

Types of Physical Interfaces

There are two sets of physical interfaces on the Net-Net 4000 SBC.

- Front interfaces are on two removable physical interface cards
- Rear interfaces are directly connected to the Net-Net 4000 SBC chassis on three 10/100 Ethernet ports

Front Interfaces

The physical interface cards installed on your Net-Net 4000 SBC determine the number of interfaces, hardware protocol, and connection speed your Net-Net 4000 SBC can use for media and signaling traffic.

- The GigE physical interface cards offer either one or two ports per card, and can use single mode or multimode fiber with an LC connector.
 - Single-port GigE card—1 Gbps of bandwidth per configured port, totaling 2 Gbps total throughput with two cards installed on the chassis.
 - Two-port GigE card—1 Gbps of bandwidth per configured port, totaling 4 Gbps total throughput with two cards installed on the chassis.
- The 10/100 Ethernet physical interface card offers four ports per card, allowing eight 10/100 Ethernet connections.

For more information about physical interface cards, including installation and cabling, refer to the *Net-Net 4000 Hardware Guide*.

Rear Interfaces

The first rear interface (wancom0) is used to carry traffic such as:

- SNMP
- Telnet
- SSH
- FTP
- ACP/XML
- Logs sent from the Net-Net SBC
- Boot the Net-Net SBC from a remote file server

The other two rear interfaces (port 1 and port 2) are used for state replication for HA. For HA, these rear interfaces on the Net-Net 4000 SBCs are directly connected by a crossover cable.

Note: To learn about HA, refer to this guide's "HA Nodes" chapter.

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

Parameter	Front Interface	Rear Interface
name	R	R
operation-type	R	R
port	R	R
slot	R	R
virtual-mac	O	I
admin-state	R	I
auto-negotiation	R	I
duplex-mode	R	I

R = Required, O = Optional, I = Invalid

Parameter	Front Interface	Rear Interface
speed	R	I
wancom-health-score	I	O

R = Required, O = Optional, I = Invalid

Before You Configure

Before you configure a physical interface:

- Decide on the number and type of physical interfaces you need.
For example, you might have one media and signaling interface connecting to a private network and one connecting to the public network. You might also need to configure a maintenance interface for HA functionality.
- If you are configuring your Net-Net 4000 SBC for HA, refer to “HA Nodes” chapter and follow the instructions there for setting special parameters in the physical interface configuration.

ACLI Instructions and Examples

This section describes how to configure the name, location, and Ethernet parameters for Net-Net 4000 SBC physical interfaces.

To add a physical interface:

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

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

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

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

Identity and State

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

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

1. **name**—Set a name for the interface using any combination of characters entered without spaces. For example: **Internet** (for a Fast Ethernet media and signaling interface) or **mai nt0** (for a maintenance interface).
2. **admin-state**—Leave the administrative state parameter set to **enabled** to receive and send media and signaling on a front interface. Select **disabled** to prevent media and signaling from being received and sent. The default for this parameter is **enabled**. The valid values are:
 - enabled | disabled

For rear interfaces, you do not need to set this parameter. Rear interfaces are always enabled.

Operation Type and Location

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

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

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

Auto-negotiation for 10/100 Front Interfaces

For 10/100 front interfaces, you need to set parameters that enable or disable auto-negotiation, set the duplex mode, and set the data rate.

ACLI Shortcut: You can set these parameters on a one-time-basis using the ACLI **set-front-interfaces** command.

Set the following parameters to configure auto-negotiation for 10/100 front interfaces:

1. **auto-negotiation**—Leave this parameter set to **enabled** so that the Net-Net SBC 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 4000 SBC begins to negotiate the link to the connected device at the duplex mode you configure. If auto-negotiation is disabled, then the Net-Net 4000 SBC will not engage in a negotiation of the link and will operate only at the duplex mode and speed you set. The default is **enabled**. The valid values are:
 - enabled | disabled
2. **duplex-mode**—Set the duplex mode. The default is **full**.
Given an operating speed of 100 Mbps, full duplex mode lets both devices on a link send and receive packets simultaneously using a total bandwidth of 200 Mbps. Given the same operating speed, half duplex mode limits the devices to one channel with a total bandwidth of 100 Mbps. The valid values are:
 - half | full
3. **speed**—Set the speed in Mbps of the front-panel 10/100 Phy interfaces; this field is only used if the auto-negotiation field is set to disabled for 10/100 Phy cards. **100** is the default. The valid values are:
 - 10 | 100

HA Configuration

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

Phy Link Redundancy

If you have two two-port GigE cards installed in your Net-Net 4000 SBC, you can configure them for phy link redundancy. This feature requires that two-port GigE cards be installed in both slots of your Net-Net 4000 SBC.

In this redundancy scheme, port 0 on slots 0 and 1 is the master port and port 1 is the backup port. The card receives and sends all traffic on one port, while the other acts as a standby in the event of failure. In this way, the two-port GigE card behaves as though it were a single-port card by only using one port as an active at one time.

You turn this feature on by setting one parameter in the system configuration; this feature is enabled for the entire system and not on a per-card basis. ACLI commands have been added so that you force a switchover from one port to another on either card, and so that you can view information about all of the ports (such as which ports are currently active, and the number of switchover events since the last reboot).

How It Works

The value of enabling this feature is that, in the event of a network or link failure, the Net-Net 4000 SBC will automatically fail over to another physical link. The Net-Net 4000 SBC polls link state on a one-second basis, so the maximum outage you might experience prior to link failure detection and switchover is one second. And if gateway heartbeats are enabled, then gateway timeout alarms will also cause failovers.

If you are not using an HA node set-up with two Net-Net 4000 SBCs, then this feature can provide link-level redundancy. Even if you are using two systems as an HA node, this feature can prevent the need for one Net-Net 4000 SBC in the node to failover to the other by simply failing over its own link; the failure of one link does not cause health score decrements that result in a system-to-system switchover. However, in the event that both the active and standby ports fail on a single slot, the Net-Net 4000 SBCs will decrement its health score so that an active-to-standby switchover will occur.

Caveats

Be aware that DoS protection and QoS metrics are not compatible with this feature. However, hostpath DoS protection is still available when you enable phy link redundancy.

The Net-Net 4000 SBC does not support this feature for any other kind of physical interface card besides the two-port Gig E card. If you have other types of cards installed in your system and try to enable this feature, the following message will display on your console and will appear in the logs:

```
Slot 1 is not a 2 Port Gigabit Card
Both Phy Interface Cards Need to be Dual Gigabit
to support the link redundancy feature
```

ACLI Instructions and Examples

This section shows you how to enable phy link redundancy, how to force a switchover, and how to view information about the redundancy links.

Note that, by default, the primary port is always port 0, and the standby port is always port 1. You should only configure port 0.

To enable phy link redundancy:

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)# system
ACMEPACKET(system)#
3. Type **system-config** and press <Enter>.
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
4. **link-redundancy-state**—Set this parameter to **enabled** if you want to use phy link redundancy for your system with two two-port GigE cards installed. A value of **disabled** turns this feature off. The default is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

To force a switchover:

1. In Superuser mode, use the new **switchover-redundancy-link** command to change the roles of the active and the standby ports by switching the active port on the slot you specify.

You carry out this command by typing **switchover-redundancy-link** and a <Space>, the slot number (0 or 1) and a <Space>, and then the port number (0 or 1). Then press <Enter>.

ACMEPACKET# **switchover-redundancy-link 0**

If the command is successful, then no further information will be displayed.

The system allows you to switch links only if the newly active link is up. If it is not, then the system displays information that tells you why the operation could not be completed:

```
Switch From Slot 1 Port 1, to Port 0 was not completed
Due to the fact Link State for Slot 1 Port 0 is down
```

Physical Interfaces: Net-Net 4500 SBC

There are two sets of physical interfaces on the Net-Net 4000 SBC. All interfaces are located on the network interface unit (NIU), which is found on the rear of the system chassis. For more information about the NIU, refer to the *Net-Net 4500 Hardware Installation Guide*.

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

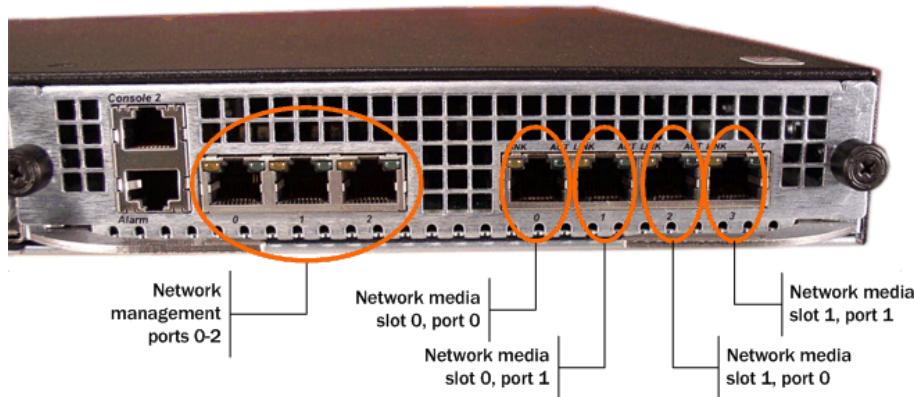
The following picture of the NIU shows you how the network media and network management ports appear. These designations are an important point of reference when you set up physical interface configurations. Note that the slot parameter for network management ports will always be set to zero (0).

Network Media Interfaces

The NIU installed on your Net-Net 4000 SBC determines the number of interfaces, hardware protocol, and connection speed your Net-Net 4000 SBC can use for media and signaling traffic.

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

For more information about NIUs, including installation and cabling procedures, refer to the *Net-Net 4500 Hardware Guide*.



Network Management Interfaces

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

- SNMP
- Telnet
- SSH
- FTP
- ACP/XML
- Logs sent from the Net-Net SBC
- Boot the Net-Net SBC from a remote file server

The other two rear interfaces (port 1 and port 2) are used for state replication for high availability (HA). For HA, these interfaces on the Net-Net 4000 SBCs are directly connected by a crossover cable.

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

Parameter	Network Media Interface	Network Management Interface
name	R	R
operation-type	R	R
port	R	R
slot	R	R
virtual-mac	O	I
admin-state	R	I
auto-negotiation	R	I

R = Required, O = Optional, I = Invalid

Parameter	Network Media Interface	Network Management Interface
duplex-mode	R	I
speed	R	I
wancom-health-score	I	O

R = Required, O = Optional, I = Invalid

Before You Configure

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

Before you configure a physical interface:

1. Decide on the number and type of physical interfaces you need.
For example, you might have one media interface connecting to a private network and one connecting to the public network. You might also need to configure maintenance interfaces for HA functionality.
2. Determine the slot and port numbering you will need to enter for the physical interfaces you want to configure. The graphic above can serve as your slot and port numbering reference.
3. If you are configuring your Net-Net 4500 SBC for HA, refer to “HA Nodes” chapter and follow the instructions there for setting special parameters in the physical interface configuration.

ACLI Instructions and Examples

You configure physical interfaces for your Net-Net 4500 SBC the same way you would for your Net-Net 4250. The only difference is that you must take care to configure the slot and port numbers that reference the NIU correctly.

For step-by-step ACLI configuration instructions, refer to the following sections in this chapter:

- [Identity and State \(104\)](#)
- [Operation Type and Location \(104\)](#)
- [Auto-negotiation for 10/100 Front Interfaces \(104\)](#)

Interface Utilization: Graceful Call Control, Monitoring, and Fault Management

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

Calculation Overview

When enabled to do so, the Net-Net SBC performs a network utilization calculation for each of its media ports. This calculation takes into account rates of receiving and transmitting data, the speed at which each is taking place, and the quality of data

traversing the interface. The Net-Net SBC keeps statistics for each media port so it can compare previously- and newly-retrieved data. For heightened accuracy, calculations are performed with milliseconds (rather than with seconds).

Alarms

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

For example, you might configure the following alarms:

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

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

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

ACLI Instructions and Examples

Configuring Utilization Thresholds for Media Interfaces

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

To configure utilization thresholds for media interfaces:

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

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

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

```
3. Type **phy-interface** and press <Enter>. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

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

```
4. Type **network-alarm-threshold** and press <Enter>.

```
ACMEPACKET(phy-interface)# network-alarm-threshold
ACMEPACKET(network-alarm-threshold)#

```
5. **severity**—Enter the severity for the alarm you want to fine for this interface: **minor** (default), **major**, or **critical**. Since the parameter defaults to minor, you must change the value if you want to define a major or critical alarm.
6. **value**—Enter the percentage of utilization (transmitting and receiving) for this interface that you want to trigger the alarm. For example, you might define a minor alarm with a utilization percentage of 50. Valid values are between 0 and 100, where 0 is the default.

7. Save your work.

Configuring Graceful Call Control

You can enable the Net-Net SBC to stop receiving session-initiating traffic on a media interface when the traffic for the interface exceeds the critical threshold you define for it.

To enable graceful call control:

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

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

```
2. Type **system** and press <Enter>.

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

```
3. Type **phy-interface** and press <Enter>. If you are adding this feature to an existing configuration, then remember you must select the configuration you want to edit.

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

```
4. **overload-protection**—Change this parameter's value to enabled if you want to turn graceful call control on. Leave it set to disabled (default) if you do not want to use this feature.
5. Save your work.

Network Interfaces

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 SBC network interface has standard parameters common to nearly all IPv4 network interfaces. There are a few fields that are unique to the Net-Net SBC.

VLANs

VLANs are used to logically separate a single physical interface into multiple network interfaces. There are several applications for this like MPLS VPNs (RFC 2547), MPLS LSPs, L2VPNs (IPSec, L2TP, ATM PVCs), reusing address space, segmenting traffic, and maximizing the bandwidth into a switch or router. The range of services and management capabilities you can implement with VPNs is huge.

The primary applications of VLANs on the Net-Net SBC are VPNs and peering. Several peering partners may terminate their connections to a Net-Net SBC on a single physical interface. VLAN tags are used to segregate and correctly route the terminated traffic. The Net-Net SBC 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 SBC 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 SBC with a Fast Ethernet or Gigabit Ethernet interface can terminate different 802.1q VLANs into separate network interfaces, each of which 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 SBC 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 are when two or more private networks with the same addressing schemes terminate on one physical interface. The problem this creates can easily be solved by using VLAN tagging. For example, two 10.x.x.x networks terminating on one Net-Net SBC network interface will obviously not work. The Net-Net SBC includes the IPv4 Address, IPv4 Subnet Mask and 802.1q VLAN tag in its Network Interface determination. This allows Net-Net SBC to directly interface to multiple VPNs with overlapping IPv4 Address space.

HIP

By default, the Net-Net SBC's FTP, ICMP, SNMP, and Telnet services cannot be accessed via the media interfaces. In order to enable these services, the Net-Net SBC includes four fields that enable administrative traffic over the media interfaces. These are collectively known as the HIP, or host-in-path functions. The HIP parameters are effectively firewall functions that open the well-known ports for specified services on media interfaces.

Configuring Network Interfaces

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

Special Considerations

Configuration changes to network interface parameters might have an impact on boot configuration parameters. After configuring the network interface, you might receive a message indicating that you could be changing boot config parameters under the following circumstances:

- A physical interface or network interface element matches the boot interface (for example, the physical port is the same as the boot port).
- The boot configuration parameters are modified, because the IPv4 address, netmask, or gateway is different from the corresponding boot configuration parameters.

You are asked if you want to continue. If you enter yes, the configuration will be saved and then the differing boot configuration parameters will be changed. If you enter no, then the configuration is not saved and the boot configuration parameters are not changed.

Configuring the first rear physical and network interface is optional because that interface, wancom0, is implicitly created by a valid bootparam configuration that specifies the boot device, IPv4 address, subnet, and gateway.

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>.

```
ACMEPACKET# config terminal
```

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

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

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

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

```
ACMEPACKET(network-interface)#

```

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

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

```
network-interface
  name          phyTEST
  sub-port-id   0
  description
  hostname      phyttest-left
  ip-address    10.0.45.4
  netmask       255.255.255.0
  gateway       10.0.45.1
  sec-gateway
  dns-ip-primary 192.168.44.55
  dns-ip-backup1
  dns-ip-backup2
  dns-domain
  hi-pi-p-list   192.168.100.101
                  192.168.100.102
                  192.168.100.103
                  192.168.100.100
  ftp-address   192.168.100.101
  icmp-address  192.168.100.102
  snmp-address
  telnet-address 192.168.100.103
ACMEPACKET(network-interface)#

```

IP Configuration and Identification

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

Set the following parameters to configure a network interface:

1. **name**—Set the name for the network interface. This must be the same name as the physical interface to which it corresponds.
2. **description**—Enter a description of the network for easier identification.
3. **hostname**—Set the hostname (FQDN) of this network interface. This parameter is optional.
4. **ip-address**—Set the IPv4 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. You can set an additional, secondary gateway via the **secondary-gateway** parameter.
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 name used to populate incomplete hostnames that do not include a domain for use with DNS queries. Entries must follow the Name format.

VLANs

One parameter is required to configure VLANs on a Net-Net SBC. 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**—Enter the identification of a specific virtual interface in a physical interface (e.g., a VLAN tab). If this network interface is not channelized, leave this field blank, and the value will correctly default to 0. The **sub-port-id** is only required if the operation type is Media. The valid range is:
 - Minimum—0
 - Maximum—4095.

HIP Addresses

To configure administrative service functionality on a media interface, you must define the IPv4 addresses on the front physical interfaces of your Net-Net SBC where you will receive administrative traffic. Adding HIP entries automatically opens the well-known port associated with a service.

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

1. **add-hip-ip**—Set all possible IPv4 address(es) on which you want the Net-Net SBC to accept administrative traffic. Entries in this element are IPv4 addresses of front panel network interfaces. This parameter can accept multiple IPv4 addresses. You can later remove this entry by typing **remove-hip-ip** followed by the appropriate IPv4 address.
2. **add-ftp-ip**—Set the IPv4 address where ports 20 and 21 are opened. This lets standard FTP packets enter the Net-Net SBC and reach the host. You can later remove this entry by typing **remove-ftp-ip** followed by the appropriate IPv4 address.
3. **add-icmp-ip**—Set the IPv4 addresses to pass standard ping packets to the host; this parameter can accommodate multiple ping IPv4 addresses. You can later remove this entry by typing **remove-icmp-ip** followed by the appropriate IPv4 address.

When you configure multiple ICMP ping addresses in for a network interface, you must also configure the host-in-path addresses in the hip-ip-list for each ICMP address. For security, if the ICMP address and the hip-ip-list are not added for an address, the Net-Net 400 hardware discards ICMP requests or responses for the address.

To remove multiple IP addresses at one time, type the **remove-icmp-ip** and a <Space>, open quotation mark ("), the IP addresses you want removed from the list each separated by a space, close quotation mark ("), and then press <Enter>.

```
ACMEPACKET (network-interface)# remove-icmp-ip "142.214.5.34
124.8.67.3"
```

4. **add-snmp-ip**—Set the IPv4 address where port 161 is opened. This lets SNMP traffic enter the Net-Net SBC and reach the host. You can later remove this entry by typing **remove-snmp-ip** followed by the appropriate IPv4 address.
5. **add-telnet-ip**—Set the IPv4 address where port 23 is opened for telnet access. You can later remove this entry by typing **remove-telnet-ip** followed by the appropriate IPv4 address.

SNMP

This section explains how to configure Simple Network Management Protocol (SNMP), trap receivers, and syslog servers. These features are not essential for baseline Net-Net SBC service, but they are necessary to use Acme Packet's Net-Net EMS to manage Net-Net SBCs. They provide important monitoring and system health information that contribute to a robust deployment of the Net-Net system.

For detailed descriptions of the MIBs and information concerning their architecture and use, please refer to the *Net-Net MIB Reference Guide*.

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 SBC. These settings are system-wide configurations including MIB contact information, SNMP community settings, and trap receivers.

Basic SNMP Parameters

The Net-Net SBC 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 SBC 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 SBC lets you define two types of access levels: read-only and read-write. You can define multiple SNMP communities on a Net-Net SBC 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 SBC. An SNMP trap is the notification sent from a network device, the Net-Net SBC in this case, that declares a change in service. Multiple trap receivers can be defined on a Net-Net SBC either for redundancy or to segregate alarms with different severity levels to individual trap receivers.

Each server that Net-Net EMS is installed on should be configured as a trap receiver on all Net-Net SBC's managed by Net-Net EMS.

Configuring SNMP

This section describes how to configure your Net-Net SBC 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 SBC. Also included here are switches that govern how the SNMP system responds to specified events.
3. Set the syslog events (explained in the next section). They can trigger SNMP syslog traps. Parameters dealing with SNMP monitoring syslog events are configured here.
4. Set SNMP communities. Their configuration is separated into a unique configuration element.
5. 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)#[/b]

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

mi b-system-contact	John Doe
mi b-system-name	Test System
mi b-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-high-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 SBC 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 SBC 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 SBC to **enabled** or **disabled**. By default, this parameter is set to **enabled**. The valid values are:
 - enabled | disabled
5. **enable-snmp-syslog-notify**—Set whether SNMP traps are sent when a Net-Net system generates a syslog message. The SNMP agent will send a trap when a syslog is generated if the following conditions are met:
 - SNMP is enabled.
 - This field is enabled.
 - The syslog severity level is equal to or greater than the severity level configured in the SNMP Syslog Level field.

The default is **disabled**. Valid values are:

- enabled | disabled

6. **enable-snmp-monitor-traps**—When this parameter is enabled, the Net-Net SBC generates traps with unique trap-IDs for each syslog event. If this parameter is disabled, a single trap-ID is used for all events, with different values in the description string. The default is **disabled**. The valid values are:
 - enabled | disabled
7. **enable-snmp-auth-traps**—Set whether the SNMP authentication traps are enabled. If an SNMP request fails authentication because of an IPv4 address and SNMP community mismatch, the SNMP request will be rejected. This field determines if an SNMP trap will be sent in response to the authentication failure. The default is **disabled**. Valid values for this parameter are:

- enabled | disabled
8. **enable-env-monitor-traps**—Set whether or not the SNMP environment monitor traps are enabled. Environment traps include main board PROM temperature, CPU voltage, power supplies, fan speeds, etc. The default is **disabled**. Valid values for this parameter are:
- enabled | disabled
9. **snmp-syslog-his-table-length**—Set the length of the syslog trap history table. When a syslog message that meets the SNMP syslog level field criteria is generated and SNMP is enabled, the SNMP agent adds that message to a history table. This parameter indicates the number of entries the table can contain. The default is **1**. The valid range is:
- Minimum—1
 - Maximum—500
- Once the last table entry is filled, the oldest entry will be overwritten with a new entry.
10. **snmp-syslog-level**—Set the log severity level threshold that will cause the syslog trap to be sent to an NMS. When this criteria is met and the appropriate SNMP trap is sent, an entry is written to the SNMP Syslog History Table. The default is **warning**. The following are valid values:
- emergency | critical | major | minor | warning | notice | info | trace | debug | detail

SNMP Community Configuration

To configure SNMP communities:

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

ACMEPACKET(system)# **snmp-community**

ACMEPACKET(snmp-community)#

From this point, you can set SNMP community parameters.

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

snmp-community	
community-name	publ i c
access-mode	READ-ONLY
i p-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 SBC 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 SBC. The default is **read-only**. The valid values are:
 - **read-only**—allows GET requests.
 - **read-write**—allows both GET and SET 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 SBC. Include the IPv4 addresses of all servers where Net-Net EMS is installed.

Trap Receiver Configuration

To configure trap receivers:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# 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
filter-level	All
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. **filter-level**—Set the filter level threshold that indicates the severity level at which a trap to be sent to this particular trap receiver. The default for this parameter is **critical**.

Example: A trap with a severity level of Critical is generated, the SNMP agent will only send this trap to NMSs that are configured in a trap-receiver element and have a filter-level parameter of Critical.

The following table maps Syslog and SNMP alarms to trap receiver filter levels.

Filter Level	Syslog Severity Level	(SNMP) Alarm Severity Level
Critical	<ul style="list-style-type: none"> • Emergency (1) • Critical (2) 	<ul style="list-style-type: none"> • Emergency • Critical
Major	<ul style="list-style-type: none"> • Emergency (1) • Critical (2) • Major (3) 	<ul style="list-style-type: none"> • Emergency • Critical • Major
Minor	<ul style="list-style-type: none"> • Emergency (1) • Critical (2) • Major (3) • Minor (4) 	<ul style="list-style-type: none"> • Emergency • Critical • Major • Minor
All	<ul style="list-style-type: none"> • Emergency (1) • Critical (2) • Major (3) • Minor (4) • Warning (5) • Notice (6) • Info (7) • Trace (8) • Debug (9) 	<ul style="list-style-type: none"> • Emergency • Critical • Major • Minor • Warning

When configuring the trap-receiver element for use with Net-Net EMS systems, Acme Packet recommends that the filter-level parameter be set to **All** for that configuration element that includes Net-Net EMS servers.

3. **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 is a critical part of diagnosing misconfigurations and optimizing operations. Net-Net SBCs can send both syslog and process log data to appropriate hosts for storage and analysis.

Overview

The Net-Net SBC generates two types of logs, syslogs and process logs. Syslogs conform to the standard used for logging servers and processes as defined in RFC 3164.

Process logs are 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. Please contact your Acme Packet sales representative directly or through email at support@acmepacket.com for more information about the process log application.

Syslog and process log servers are both identified by an IPv4 address and port pair.

Process Log Messages

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

<file-name>: <log-message>

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

Syslog and Process Logs Configuration

ACLI Instructions and Examples

This section describes how to configure syslog and process log servers.

To configure syslogs and process logs:

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

```
ACMEPACKET# config terminal
```

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

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

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

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

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

From this point, you can set process log parameters. Skip to the following process log configuration section.

4. Type **syslog-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual syslog parameters

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

```
ACMEPACKET(syslog-server)#
```

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

system-log-level	WARNING
syslog-server	
address	172.15.44.12
port	514
facility	4
process-log-level	NOTICE
process-log-ip-address	0.0.0.0
process-log-port	0

Syslog Configuration

The Net-Net SBC supports multiple syslog servers. As the number of active syslog increases, the performance level of the Net-Net SBC may decrease. Therefore, we recommend configuring no more than 8 syslog servers.

Set the following parameters to configure syslog servers:

1. **address**—Set the IPv4 address of a syslog server.
2. **port**—Set the port portion of the syslog server. The default is **514**.

3. **facility**—Set an integer to identify a user-defined facility value sent in every syslog message from the Net-Net SBC to the syslog server. This parameter is used only for identifying the source of this syslog message as coming from the Net-Net SBC. It is not identifying an OS daemon or process. The default value for this parameter is 4. RFC 3164 specifies valid facility values.

In software release versions prior to Release 1.2, the Net-Net SBC would send all syslog messages with a facility marker of 4.

4. **system-log-level**—Set which log severity levels write to the system log (filename: acmelog). The default is **WARNING**. Valid values are:
 - EMERGENCY | CRITICAL | MAJOR | MINOR | WARNING | NOTICE | INFO | TRACE | DEBUG | DETAIL

Process Log Configuration

Set the following parameters to configure the process log server:

1. **process-log-level**—Set the starting log level all processes running on the Net-Net system use. Each individual process running on the system has its own process log. The default is **NOTICE**. Valid values are:
 - EMERGENCY | CRITICAL | MAJOR | MINOR | WARNING | NOTICE | INFO | TRACE | DEBUG | DETAIL
2. **process-log-ip-address**—Set the IPv4 address of the process log server. The default **0.0.0.0**, which causes log messages to be written to the normal log file.
3. **process-log-port**—Set the port number associated with the process log server. The default value for this parameter is **0**, which causes log messages to be written to the normal log file. The valid range is:
 - Minimum—0
 - Maximum—65535.

Host Routes

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

Overview

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

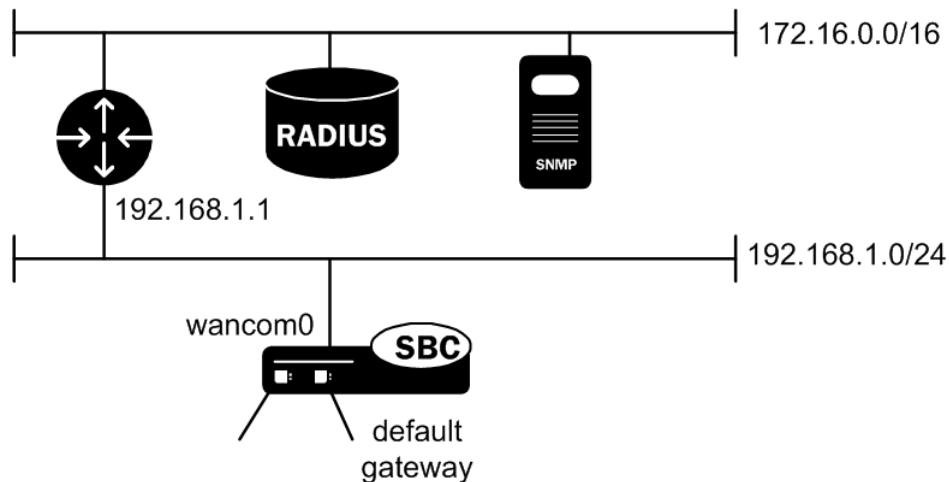
When traffic is destined for a network that is not explicitly defined on a Net-Net SBC, the default gateway (located in the system config) is used. If you try to route traffic to a specific destination that is not accessible through the default gateway, you need to add a host route. Host routes can be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

Certain SIP configurations require that the default gateway is located on a front media interface. In this scenario, if management applications are located on a network connected to a rear-interface network, you will need to add a host route for management connectivity.

When source-based routing is used, the default gateway must exist on a front media interface. Host routes might be needed to reach management applications connected to a wancom port in this kind of situation as well.

Host Routes Example

Because SIP signaling over media interfaces is enabled, the default gateway uses an IPv4 address assigned to a media interface. Maintenance services (SNMP and Radius) are located on a network connected to, but separate from, the 192.168.1.0/24 network on wancom0. In order to route Radius or SNMP traffic to an NMS (labeled as SNMP in the following example), a host route entry must be a part of the Net-Net SBC configuration. The host route tells the host how to reach the 172.16.0.0/16 network. The actual configuration is shown in the example in the next section of this guide.

**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# **config 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. Parameters not described in this section are omitted below.

```

host-routes
  dest-network      172.16.0.0
  netmask          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. **dest-network**—Set the IPv4 address of the destination network that this host route points toward.
2. **netmask**—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.

Host routes can now be configured as an RTC-enabled configuration element. You only need to add, change, or delete a host route followed by a **save-config** and then **activate-config** in order to change the host route configuration. You do not need to reboot your Net-Net SBC to complete a host route change.

Holidays

This section explains how to configure holidays on the Net-Net SBC.

You can define holidays that the Net-Net SBC recognizes. Holidays are used to identify a class of days on which a local policy is enacted. All configured holidays are referenced in the **local-policy-attributes** configuration subelement as an H in the **days-of-week** parameter. Because holidays are entered on a one-time basis per year, you must configure a new set of holidays yearly.

ACLI Instructions and Examples

To configure holidays:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# session-router
3. Type **session-router-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# session-router-config
ACMEPACKET(session-router-config)#
4. Type **holidays** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router-config)# holidays
ACMEPACKET(session-router-holidays)#

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

holiday

date	2005-01-01
description	New Years Day

To configure a holiday, add an entry for the following parameters in the **holidays** element:

1. **date**—Enter the holiday's date in YYYY-MM-DD format.

2. **description**—Enter a short description for the holiday you are configuring. If the description contains words separated by spaces, enter the full description surrounded by quotation marks.

Enhanced Control of UDP and TCP Ports

This section explains how to configure the Net-Net SBC for finer control of the set of UDP and TCP ports that on which the Net-Net SBC provides services. The settings you can configure have an impact on:

- UDP/TCP port 111 (the RPC services port), which is disabled on Net-Net SBC startup but can be enabled in the boot parameters
- TCP ports 3000 (used when notify commands are issued remotely, i.e. via the Net-Net EMS) and 3001 (used for remote configuration, i.e. via the Net-Net EMS), which can now be enabled or disabled in the system configuration

Neither configuration for these features is covered by RTC, so you must reboot your Net-Net SBC for changes to take effect. Be aware that rebooting can cause system downtime, and plan accordingly.

ACLI Instructions and Examples

To enable port 111 using Net-Net SBC boot parameters:

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# **config terminal**
2. To enter the boot parameters so that you can configure them, type **bootparam** and press <Enter>.
ACMEPACKET(config)# **bootparam**
3. Press <Enter> to scroll through the list of boot parameters until you reach the setting for flags.

To set this value correctly, you need to add the value **0x200000** to your existing flag setting in the boot parameters. In the example below, the existing flag value is **0x30008**. When the value **0x200000** is added, the result is **0x230008**. The result is the value that you need to set.

When you reach the **flag** setting, type the value representing the flags you need (**0x230008** in the example below) and press <Enter>. Continue to press <Enter> to finish scrolling through the rest of the boot parameters.

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

```
boot device          : wancom0
processor number : 0
host name : acmepacket8
file name : /tffs0/sd220p9.gz
inet on ethernet (e): 10.0.1.57:fffff0000
inet on backplane (b): 0.0.0.0
host inet (h)      : 10.0.1.5
gateway inet (g)   : 10.0.0.1
user (u)           : user
ftp password (pw)  : password
flags (f)          : 0x30008 0x230008
target name (tn)   : acmesystem
startup script (s) : 0
other (o)          :
```

NOTE: These changed parameters will not go into effect until reboot. Also, be aware that some boot parameters may also be changed through the PHY and Network Interface Configurations.

ACMEPACKET(configure)#

4. Type **exit** to return to the main Superuser menu so that you can reboot your Net-Net SBC and apply the settings you have entered.

ACMEPACKET(configure)#
exit

5. Reboot your Net-Net SBC. Type a **y** and press <Enter> to reboot.

ACMEPACKET#
reboot

WARNING: you are about to reboot this SD!

Reboot this SD [y/n]?: y

To control TCP ports 3000 and 3001 in the system configuration:

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

ACMEPACKET#
configure terminal

2. Type **system** and press <Enter>.

ACMEPACKET(configure)#
system

3. To set parameters in the system configuration, type **system-config** and press <Enter>.

ACMEPACKET(system)#
system-config

4. To start editing the system configuration, type **select** and press <Enter>.

ACMEPACKET(system-config)#
select

5. The parameter controlling ports 3000 and 3001 is called **remote-control**, and its default is enabled. To disable the ports, set this parameter to disabled.

ACMEPACKET(system-config)#
remote-control disabled

6. Save your changes and exit the system configuration and main system menus.

ACMEPACKET(system-config)#
done

ACMEPACKET(system)#
exit

7. Type **exit** to return to the main Superuser menu so that you can reboot your Net-Net SBC and apply the settings you have entered.

ACMEPACKET(configure)#
exit

8. Reboot your Net-Net SBC. Type a **y** and press <Enter> to reboot.

ACMEPACKET#
reboot

WARNING: you are about to reboot this SD!

Reboot this SD [y/n]?: y

DNS Transaction Timeout

This section explains how to configure the DNS transaction timeout interval on a per network-interface basis. You can currently configure the Net-Net SBC with a primary and two optional backup DNS servers. The Net-Net SBC queries the primary DNS server and upon not receiving a response within the configured number of seconds, queries the backup1 DNS server and if that times out as well, then contacts the backup2 DNS server.

Retransmission Logic

The retransmission of DNS queries is controlled by three timers. These timers are derived from the configured DNS timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the 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
```

Configuring DNS Transaction Timeout

You can configure DNS transaction timeout using the ACLI.

ACLI Instructions and Examples

To configure DNS transaction timeout:

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

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

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

```
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)#
From this point, you can configure network interface parameters. To view all network interface parameters, enter a ? at the system prompt.
```
4. **dns-timeout**—Enter the total time in seconds you want to elapse before a query (and its retransmissions) sent to a DNS server would timeout. The default is 11 seconds. The valid range is:
 - Minimum—1
 - Maximum—999999999.

If a query sent to the primary DNS server times out, the backup1 DNS server is queried. If the query times out after the same period of time elapses, the query continues on to the backup2 DNS server.
5. Save and activate your configuration.

Persistent Protocol Tracing

This section explains how to configure persistent protocol tracing to capture specific SIP and MGCP protocol message logs and persistently send them off the Net-Net SBC, even after rebooting the system. This feature is not applicable to log for H.323 or IWF.

About Persistent Protocol Tracing

You can configure sending protocol message logs off of the Net-Net SBC, and have that persist after a reboot. You no longer have to manually issue the **notify** command each time you reboot.

To support persistent protocol tracing, you configure the following system-config parameters:

- **call-trace**—Enable/disable protocol message tracing (currently only **sipmsg.log** and **alg.log**) regardless of the process-log-level setting. If the process-log-level is set to trace or debug, call-trace will not disable.
- **internal-trace**—Enable/disable internal ACP message tracing for all processes, regardless of process-log-level setting. This applies to all *.log (internal ACP message exchange) files other than **sipmsg.log** and **alg.log**. If the process-log-level is set to trace or debug, call-trace will not disable.
- **log-filter**—Determine what combination of protocol traces and logs are sent to the log server defined by the process-log-ip parameter value. You can also “fork” the traces and logs, meaning that you keep trace and log information in local storage as well as sending it to the server. You can set this parameter to any of the following values: **none**, **traces**, **traces-fork**, **logs**, **logs-fork**, **all**, or **all-fork**.

The Net-Net SBC uses the value of this parameter in conjunction with the process-log-ip and process-log-port values to determine what information to send. If you have configured the proc-log-ip and proc-log-port parameters,

choosing **traces** sends just the trace information (provided they are turned on), **logs** sends only process logs (**log.***), and **all** sends everything (which is the default).

About the Logs

When you configure persistent protocol tracing, you affect the following types of logs. See the *Net-Net 4000 Maintenance and Troubleshooting Guide* for more details about all Net-Net SBC logs.

Note: Enabling logs can have an impact on Net-Net SBC performance.

Process Logs

Events are logged to a process log flow from tasks and are specific to a single process running on the Net-Net SBC. By default they are placed into individual files associated with each process with the following name format:

log. <taskname>

By setting the new **log-filter** parameter, you can have the logs sent to a remote log server (if configured). If you set **log-filter** to **logs** or **all**, the logs are sent to the log server. Otherwise, the logs are still captured at the level the **process-log-level** parameter is set to, but the results are stored on the Net-Net SBC's local storage.

Communication Logs

These are the communication logs between processes and system management. The logs are usually named **<name>.log**, with **<name>** being the process name. For example, **sipd.log**.

This class of log is configured by the new **internal-trace** parameter.

Protocol Trace Logs

The only protocol trace logs included at this time are **sipmsg.log** for SIP and **alg.log** for MGCP. (The H.323 system tracing is not currently included.) All of the logs enabled with the **call-trace** parameter are sent to remote log servers, if you also set the **log-filter** parameter to **logs** or **all**.

ACLI Instructions and Examples

Enabling Persistent Protocol Tracing

To configure persistent protocol tracing:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# system
3. Type **system-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
4. **call-trace**—Set to **enabled** to enable protocol message tracing for **sipmsg.log** for SIP and **alg.log** for MGCP. The default is **disabled**. The valid values are:

- enabled | disabled
5. **internal-trace**—Set to **enabled** to enable internal ACP message tracing for all processes. The default is **disabled**. The valid values are:
 - enabled | disabled
 6. **log-filter**—Choose the appropriate setting for how you want to send and/or store trace information and process logs. The valid values are:
 - **none**—No information will be sent or stored.
 - **traces**—Sends the trace information to both the log server; includes <name>.log files that contain information about the Net-Net SBC's internal communication processes (<name> is the name of the internal process)
 - **traces-fork**—Sends the trace information to both the log server and also keeps it in local storage; includes <name>.log files that contain information about the Net-Net SBC's internal communication processes (<name> is the name of the internal process)
 - **logs**—Sends the process logs to both the log server; includes log.* files, which are Net-Net SBC process logs
 - **logs-fork**—Sends the process logs to both the log server and also keeps it in local storage; includes log.* files, which are Net-Net SBC process logs
 - **all**—Sends all logs to the log servers that you configure
 - **all-fork**—Sends all logs to the log servers that you configure, and it also keeps the logs in local storage
 7. Save and activate your configuration.

System Access Control

You can configure a system access control list (ACL) for your Net-Net SBC that determines what traffic the Net-Net SBC allows over its management interface (wancom0). 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, however, that all IP addresses configured in the SNMP community table are automatically permitted.

ACL Instructions and Examples

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

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

Adding an ACL for the Management Interface

You access the **system-access-list** via system path, where you set an IP address and netmask. You can configure multiple system ACLs using this configuration.

To add an ACL for the management interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **system** and press <Enter> to access the signaling-level configuration elements.
ACMEPACKET(configure)# **system**
ACMEPACKET(system)#
 3. Type **system-access-list** and press <Enter>.
ACMEPACKET(system)# **system-access-list**
ACMEPACKET(system-access-list)#
 4. **source-address**—Enter the IP address representing for the source network for which you want to allow traffic over the management interface.
 5. **netmask**—Enter the netmask portion of the source network for the traffic you want to allow. The netmask is in dotted decimal notation.

Notes on Deleting System ACLs

If you delete a system ACL from your configuration, the Net-Net SBC checks whether or not there are any active FTP or Telnet client was granted access when the entry was being removed. If such a client were active during ACL removal, the 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 suspended.

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 connections 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 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-interface basis. For example, the H.323 interface (stack) configuration allows you to enable or disabled use of the network parameters settings.

ACLI Instructions and Examples

TCP 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 take Steps 1 through 4. You can simply set the TCP keepalive function in the H.323 stack configuration, and the defaults for network parameters will be applied.

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure terminal
2. Type **system** and press <Enter> to access the system-related configurations.
ACMEPACKET(configure)# system
3. Type **network-parameters** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
4. **tcp-keepinit-timer**—If a TCP connection cannot be established within some amount of time, TCP will time out the connect attempt. It can be used to set the initial timeout period for a given socket, and specifies the number of seconds to wait before the connect attempt is timed out. For passive connections, this value is inherited from the listening socket. The default is 75. The valid range is:
 - Minimum—0
 - Maximum—999999999.
5. **tcp-keepalive-count**—Enter the number of packets the 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

Configurable TCP Timers

You can configure your Net-Net SBC to detect failed TCP connections more quickly so that data can be transmitted via an alternate connection before timers expire. Across all protocols, you can now control the following for TCP:

- Connection establishment
- Data retransmission
- Timer for idle connections

These capabilities all involve configuring an **options** parameter that appears in the network parameters configuration.

ACLI Instructions and Examples

Configuring TCP Connection Establishment

This section explains the details about each facet of the configurable TCP timers feature and how to configure each.

To establish connections, TCP uses a three-way handshake during which two peers exchange TCP SYN messages to request and confirm the active open connection. In attempting this connection, one peer retransmits the SYN messages for a defined period of time if it does not receive acknowledgement from the terminating peer. You can configure the amount of time in seconds between the retries as well as how long (in seconds) the peer will keep retransmitting the messages.

You set two new options in the network parameters configuration to specify these amounts of time: **atcp-syn-rxmt-interval** and **atcp-syn-rxmt-maxtime**.

Note that for all configured options, any values entered outside of the valid range are silently ignored during configuration and generate a log when you enter the **activate** command.

To configure TCP connection establishment:

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

```
ACMEPACKET# config terminal
```

2. Type **system** and press <Enter>.

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

```

3. Type **network-parameters** and press <Enter>.

```
ACMEPACKET(system)#
ACMEPACKET(network-parameters)#

```

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **atcp-syn-rxmt-interval=x** (where x is a value in seconds between 2 and 10) with a “plus” sign in front of it. Then press <Enter>. This value will be used as the interval between TCP SYN messages when the Net-Net SBC is trying to establish a connection with a remote peer.

Now enter a second option to set the maximum time for trying to establish a TCP connection. Set the options parameter by typing **options**, a <Space>, the option name **atcp-syn-rxmt-maxtime=x** (where x is a value in seconds between 5 and 75) with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(network-parameters)# options +atcp-syn-rxmt-interval =5
ACMEPACKET(network-parameters)# options +atcp-syn-rxmt-maxtime=30
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

Note that the **atcp-syn-rxmt-maxtime=x** option is equivalent to the **tcp-keepinit-timer** parameter, but only affects ATCP.

5. Save and activate your configuration.

Configuring TCP Data Retransmission

TCP is considered reliable in part because it requires that entities receiving data must acknowledge transmitted segments. If data segments go unacknowledged, then they are retransmitted until they are finally acknowledged or until the maximum number of retries has been reached. You can control both the number of times the Net-Net SBC tries to retransmit unacknowledged segments and the periodic interval (how often) at which retransmissions occur.

You set two new options in the network parameters configuration to specify how many retransmissions are allowed and for how long: **atcp-rxmt-interval** and **atcp-rxmt-count**.

To configure TCP data retransmission:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **system** and press <Enter>.
ACMEPACKET(configure)# **system**
ACMEPACKET(system)#
3. Type **network-parameters** and press <Enter>.
ACMEPACKET(system)# **network-parameters**
ACMEPACKET(network-parameters)#
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **atcp-rxmt-interval=x** (where x is a value in seconds between 2 and 60) with a “plus” sign in front of it. Then press <Enter>. This value will be used as the interval between retransmission of TCP data segments that have not been acknowledged.

Now enter a second option to set the number of times the Net-Net SBC will retransmit a data segment before it declares the connection failed. Set the options parameter by typing **options**, a <Space>, the option name **atcp-rxmt-count=x** (where x is a value between 4 and 12 representing how many retransmissions you want to enable) with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(network-parameters)# options +atcp-rxmt-interval =30
```

```
ACMEPACKET(network-parameters)# options +atcp-rxmt-count=6
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Timer for Idle Connections

When enabled to do so, the Net-Net SBC monitors inbound TCP connections for inactivity. These are inbound connections that the remote peer initiated, meaning that the remote peer sent the first SYN message. You can configure a timer that sets the maximum amount of idle time for a connection before the Net-Net SBC consider the connection inactive. Once the timer expires and the connection is deemed inactive, the Net-Net SBC sends a TCP RST message to the remote peer.

To configure the timer for TCP idle connections:

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

```
ACMEPACKET# configure terminal
```

2. Type **system** and press <Enter>.

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

```
ACMEPACKET(system)#
```

3. Type **network-parameters** and press <Enter>.

```
ACMEPACKET(system)# network-parameters
```

```
ACMEPACKET(network-parameters)#{}
```

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **atcp-idle-timer=x** (where x is a value in seconds between 120 and 7200) with a “plus” sign in front of it. Then press <Enter>. This value will be used to measure the activity of TCP connections; when the inactivity on a TCP connection reaches this value in seconds, the Net-Net SBC declares it inactive and drops the session.

```
ACMEPACKET(network-parameters)# options +atcp-idle-timer=900
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

Historical Data Recording (HDR)

Historical data recording (HDR) refers to a group of management features that allow you to configure the Net-Net SBC to collect statistics about system operation and function, and then send those records to designated servers. System statistics, defined in detail below, are saved to a comma-separated value (CSV) file, which are then sent to the designated server(s).

Information types are grouped so that you can refer to a set of statistics by simply invoking their group name. Within each group, there are several metrics available.

How It Works

In the system configuration, you can enable HDR by first turning on the system's collection function, then choosing the records you want to capture, and finally setting up server(s) to which you want records sent.

The main collect configuration (found within the main system configuration) allows you to create global settings that:

- Enable or disabled HDR at boot time
- Set the sample rate in seconds, or the time between sample individual collections
- Set the time in seconds in between individual pushes to designated servers (configured in the push receiver configuration accessed via the collect configuration)
- Set the time you want the collect to start and stop; time is entered in year, month, day, hours, minutes, and seconds

You also configure setting for each group of data you want to collect, and the push receiver (server) to which you want data sent.

Protocol Use

You can configure HDR to send files using FTP or (for added security) SFTP, but FTP is the default. Note that public key authentication is not available when you choose SFTP. Instead, the Net-Net SBC uses password authentication.

About the CSV File

When you enable HDR and configure one or more servers to which you want records sent, data is transmitted in a CSV file in standard format. There is one CSV file per record group type, and the first record for each file is a header containing the field name for each attribute in that file.

Collection Interval and Push

In your HDR configuration, you set parameters that govern:

- The groups for which the Net-Net SBC collects records
- How frequently the Net-Net SBC collects records
- How frequently the Net-Net SBC sends records off-box

Factoring in the number of groups for which you collect records, you can calculate the number of records that will be sent per push. The number of files that are sent off-box equals the number of groups for which the Net-Net SBC is collecting records; there is always one additional record for each group, a header file containing the field name for each attribute.

The number of records in a file, then, equals the push interval divided by the sample interval time multiplied by the number of groups, plus one. Take the case, for

example, where you set a push interval time of 60 seconds and a sample interval time of 5 seconds, with a group of ten records. With these settings, the Net-Net SBC would send 120 group records and 10 header records (for a total of 130 records) for each push.

You can configure an option parameter (disabled by default) that instructs the Net-Net SBC to send a trap when data has been successfully pushed. This trap is defined in the `ap-smgmt.mi.b` and has a default level of MINOR. It contains the name of the node that successfully pushed the HDR file to an HDR server, a unique file name for the HDR file that was pushed, and the IP address of the push receiver (configured in the global collection configuration).

Note that after each push, the Net-Net SBC clears (deletes) all records. The Net-Net SBC also clears files on system reboot, and after three consecutive push failures.

Group Record Types

In the group-name parameter for the group-settings configuration, you can enter any one of the groups record type defined in the following table. You specify the collection object, and then all metrics for that groups are sent.

Collection Object	Metrics Included
General system statistics (system)	<ul style="list-style-type: none"> • CPU Utilization • Memory Utilization • Health Score • Redundancy State • Signaling Sessions • Signaling Rate (CPS) • CAM Utilization (NAT) • Cam Utilization (ARP) • I2C Bus State • License Capacity • Current Cached SIP Local Contact Registrations • Current H323 Number of Registrations • Current MGCP Public Endpoint Gateway Registrations
Interface statistics (interface)	<ul style="list-style-type: none"> • Index • Description • Type • MTU • Speed • Physical Address • Admin Status • Operational State • IfLastChange • InOctets • InUnicastPackets • InNon-UnicastPackets • InDiscards • OutErrors • OutOctets • OutUnicastPackets • OutNon-UnicastPackets • OutDiscards • InErrors

Collection Object	Metrics Included
Combined session agent statistics (session-agent)	<ul style="list-style-type: none"> • Hostname • System Type • Status • Inbound Active Sessions • Inbound Session Rate • Outbound Active Sessions • Outbound Session Rate • Inbound Sessions Admitted • Inbound Sessions Not Admitted • Inbound Concurrent Sessions High • Inbound Average Session Rate • Outbound Sessions Admitted • Outbound Sessions Not Admitted • Outbound Concurrent Sessions High • Outbound Average Sessions Rate • Max Burst Rate • Total Seizures • Total Answered Sessions • Answer/Seizure Ratio • Average One-Way Signaling Latency • Maximum One-Way Signaling Latency
Session realm statistics (session-realm)	<ul style="list-style-type: none"> • Realm Name • Inbound Active Sessions • Inbound Session Rate • Outbound Active Sessions • Outbound Session Rate • Inbound Sessions Admitted • Inbound Sessions Not Admitted • Inbound Concurrent Sessions High • Inbound Average Session Rate • Outbound Sessions Admitted • Outbound Sessions Not Admitted • Outbound Concurrent Sessions High • Outbound Average Sessions Rate • Max Burst Rate • Total Seizures • Total Answered Sessions • Answer/Seizure Ratio • Average One-Way Signaling Latency • Maximum One-Way Signaling Latency • Average QoS RFactor • Current QoS Critical Exceeded • Current QoS Major Exceeded • Maximum QoS RFactor • Total QoS Critical Exceeded • Total QoS Major Exceeded
Environmental voltage statistics (voltage)	<ul style="list-style-type: none"> • Voltage type • Description • Current voltage (mv)
Environmental fan statistics (fan)	<ul style="list-style-type: none"> • Location • Description • Speed (% of range)
Environmental temperature statistics (temperature)	<ul style="list-style-type: none"> • Type • Description • Temperature (Celsius)

Collection Object	Metrics Included
SIP status statistics (sip-sessions)	<ul style="list-style-type: none"> • Sessions • Sessions Initial • Sessions Early • Sessions Established • Sessions Terminated • Dialogs • Dialogs Early • Dialogs Confirmed • Dialogs Terminated
SIP error/event statistics (sip-errors)	<ul style="list-style-type: none"> • SDP Offer Errors • SDP Answer Errors • Drop Media Errors • Transaction Errors • Application Errors • Media Exp Events • Early Media Exps • Exp Media Drops • Expired Sessions • Multiple OK Drops • Multiple OK Terms • Media Failure Drops • Non-ACK 2xx Drops • Invalid Requests • Invalid Responses • Invalid Messages • CAC Session Drops • CAC BW Drops
SIP INVITE	<ul style="list-style-type: none"> • Requests • Retransmissions • 100 Trying • 180 Ringing • 181 Forwarded • 183 Progress • 200 OK • 30x Moved • 403 Forbidden • 404 Not found • 408 Request timeout • 480 Unavailable • 484 Address incompl • 487 Terminated • 500 Internal error • 503 Service unavail • Response retransmissions • Transaction timeouts • Locally throttled

Collection Object	Metrics Included
SIP policy/routing (sip-policy)	<ul style="list-style-type: none"> • Local Lookup • Local Hits • Local Misses • Local Drops • Agent Group Hits • Agent Group Misses • No Routes Found • Missing Dialog • Inb SA Constraints • Outb SA Constraints • Inb REG SA Constraints • Outb REG SA Constraints • Requests Challenged • Challenges Found • Challenges Not Found • Challenge Drops
SIP server transaction (sip-server)	<ul style="list-style-type: none"> • All States • Initial • Trying • Proceeding • Cancelled • Established • Completed • Confirmed • Terminated
SIP client transactions (sip-client)	<ul style="list-style-type: none"> • All States • Initial • Trying • Calling • Proceeding • Cancelled • EarlyMedia • Completed • SetMedia • Established • Terminated
SIP ACL status (sip-ACL-status)	<ul style="list-style-type: none"> • Total Entries • Trusted • Blocked
SIP ACL operations (sip-ACL-oper)	<ul style="list-style-type: none"> • ACL Requests • Bad Messages • Promotions • Demotions

Collection Object	Metrics Included
SIP session status (sip-status)	<ul style="list-style-type: none"> • Sessions • Subscriptions • Dialogs • CallID Maps • Rejections • ReINVITEs • Media Sessions • Media Pending • Client Trans • Server Trans • Resp Contexts • Saved Contexts • Sockets • Req Drops • DNS Trans • DNS Sockets • DNS Results • Session Rate • Load Rate
MGCP task state (mgcp-state)	<ul style="list-style-type: none"> • MGCP Sessions • CA Endpoints • GW Endpoints • Media Sessions • Client Trans • Server Trans • Pending MBCD • MGCP ALGs • Port Maps Allocated • Port Maps Available
MGCP transactions (mgcp-trans)	<ul style="list-style-type: none"> • Requests Received • Responses Sent • Duplicates Received • Requests Sent • Responses Received • Retransmissions Sent
MGCP media events (mgcp-media-events)	<ul style="list-style-type: none"> • Calling SDP Errors • Called SDP Errors • Drop Media Errors • Transaction Errors • Application Errors • Media Exp Events • Early Media Exps • Exp Media Drops
MGCP ACL status (mgcp-ACL)	<ul style="list-style-type: none"> • Total Entries • Trusted • Blocked
ACL operation (mgcp-oper)	<ul style="list-style-type: none"> • ACL requests • Bad messages • Promotions • Demotions
ENUM	<ul style="list-style-type: none"> • Total queries • Successful queries • Not found queries

Collection Object	Metrics Included
H.323 statistics (h323-stats)	<ul style="list-style-type: none"> • Incoming Calls • Outgoing Calls • Connected Calls • Incoming Channels • Outgoing Channels • Contexts • Queued Messegues • TPKT Channels • UDP Channels
Space (Data for the storage expansion module)	<ul style="list-style-type: none"> • TimeStamp • Partition • Space used • Space available

ACLI Instructions and Examples

This section shows you how to configure HDR. You need to set up:

- The collection configuration to govern sample and push intervals, start and end times for collection
- Setting to support this feature across an HA node
- The group settings configuration that tells the Ne-Net 4000 what groups of records to collect, when to start and stop collecting them, and how often to sample for that group
- Push receivers that take the records the Net-Net SBC sends

All HDR parameters are RTC-supported, so you can save and activate your configuration for them to take effect.

Accessing the HDR Configuration Parameters

You access the parameters that enable and support HDR using the ACLI system-config path.

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure **term**inal
2. Type **system** and press <Enter>.
ACMEPACKET(config)# system
ACMEPACKET(system)#
3. Type **system-config** and press <Enter>.
ACMEPACKET(system)# **system-confi**g
ACMEPACKET(system-confi)#
4. Enter **collect** and press <Enter>. From here, you can type a question mark (?) to see individual parameters for the configuration.
ACMEPACKET(system)# **col**lect
ACMEPACKET(collect)#

Global Collection Settings

You access the collection configuration through the ACLI system-configuration menu. Once in the collection configuration, you can establish the global settings for HDR collection.

To configure global settings for HDR support:

1. **boot-state**—Set this parameter to **enabled** to start group collection, or to **disabled** to prevent the Net-Net SBC from collecting HDR statistics. This parameter does not go into effect until the system is rebooted. You can also use the ACLI request collect start command to start collection; using this command, you can start collection for all groups, or for one specified group. The default is **disabled**. The valid values are:
 - enabled | disabled
2. **protocol**—You only need to set this parameter if you want to use SFTP to send HDR collection record files. When you use SFTP, the Net-Net SBC uses password rather than public key authentication. The valid values are:
 - FTP (default) | SFTP
3. **sample-interval**—Enter the time in minutes for how often you want the Net-Net SBC to sample data records. The default is **0**; leaving this parameter set to **0** turns off the feature. The valid range is:
 - Minimum—0
 - Maximum—120
4. **push-interval**—Enter the time in minutes for how often you want the Net-Net SBC to send collected records to push receiver(s). The default is **0**. The valid range is:
 - Minimum—0
 - Maximum—120
5. **start-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SBC to start HDR collection; this time is either **now** or a time in the future. Your entry must be in the format **yyyy-mm-dd-hh: mm: ss**, where: **yyyy** is the year, **mm** is the month, **dd** is the day, **hh** in the hour, **mm** is the minutes, and **ss** is the second. The default is **now**.
6. **end-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SBC to finish HDR collection; this time is either **never** or a time in the future. Your entry must be in the format **yyyy-mm-dd-hh: mm: ss**, where: **yyyy** is the year, **mm** is the month, **dd** is the day, **hh** in the hour, **mm** is the minutes, and **ss** is the second. There is no default for this parameter. The default is **never**.
 1. **push-success-trap-state**—Set this parameter to **enabled** if you want the Net-Net SBC to send a trap confirming successful data pushes to HDR servers. This parameter is **disabled** by default. The valid values are:
 - enabled | disabled

HDR for an HA Node

If you are using the HDR feature on an HA node (or redundant pair of Net-Net SBCs), then you need to make sure that several parameters in the collection configuration are set appropriately.

Acme Packet recommends strongly that you do not change these parameters from their defaults for a normal HA node configuration. Therefore, if you need to change them to support HDR, you should do so with caution.

To configure parameters for HDR support across an HA node:

1. **red-collect-state**—Set the state of HA support for the collector function. The default is **disabled**. The valid values are:

- enabled | disabled
2. **red-max-trans**—Enter the maximum number of HA synchronized transactions to maintain on the active system in the HA node. The default is **1000**. The valid range is:
 - Minimum—0
 - Maximum—999999999
 3. **red-sync-start-time**—Enter the amount of time in milliseconds that the active Net-Net SBC checks to confirm that it is still the active system in the HA node. If the active system is still adequately healthy, this timer will simply reset itself. If for any reason the active has become the standby, it will start to checkpoint with the newly active system when this timer expires. The default is **5000**. The valid range is:
 - Minimum—0
 - Maximum—999999999
 4. **red-sync-comp-time**—Enter amount of time in milliseconds that determines how frequently after synchronization the standby Net-Net SBC checkpoints with the active Net-Net SBC. The first interval occurs after initial synchronizations of the systems; this is the timeout for subsequent synchronization requests. The default is **1000**. The valid range is:
 - Minimum—0
 - Maximum—999999999

Collection Group Settings

You can configure multiple collection groups on your Net-Net SBC; the names of these groups appear in the [Group Record Types \(138\)](#) section above. Collection group settings are accessible through the collection configuration.

Note that the sample collection interval, start time, and end time you set here override the ones established in the global collection settings. The largest value you can enter for an group's sample collection must be smaller than the global push interval value.

To configure collection group settings:

1. Access the collection group (**group-settings**) configuration by way of the collection configuration. Once


```
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)# group-settings
```
2. **group-name**—Enter the group name corresponding to the records that you want to collect; there are 21 possible groups for which the Net-Net SBC can collect data. The **system** group name is the default for this parameter; the other possible names to which you can refer are listed in the [Group Record Types \(138\)](#) table above.
3. **sample-interval**—Enter the time in minutes for how often you want the Net-Net SBC to sample data records for the specified group. The default is **0**; this value turns off the feature for this group. The valid range is:
 - Minimum—0
 - Maximum—120
4. **start-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SBC to start collecting records for this group; this time is either **now** or a time in the future. Your entry must be in the format **yyyy-mm-**

dd-hh: mm: ss, where: **yyyy** is the year, **mm** is the month, **dd** is the day, **hh** in the hour, **mm** is the minutes, and **ss** is the second. There is no default for this parameter.

5. **end-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SBC to stop collecting records for this group; this time is either **never** or a time in the future. Your entry must be in the format **yyyy-mm-dd-hh: mm: ss**, where: **yyyy** is the year, **mm** is the month, **dd** is the day, **hh** in the hour, **mm** is the minutes, and **ss** is the second. There is no default for this parameter.

Push Receiver Settings

You can configure multiple servers to receive the records that the Net-Net SBC. Push receiver settings are accessible through the collection configuration.

If you configure more than one server, then the Net-Net SBC sends data to all of the servers. If one server fails, the Net-Net SBC generates an SNMP trap. In terms of clearing data, this means that if there are four servers configured and the Net-Net SBC successfully pushes data to three of them, then it will clear the data.

To configure servers to act as push receivers for HDR data:

1. Access the push receiver (**push-receiver**) configuration by way of the collection configuration.

```
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)# push-receiver
```
2. **address**—Enter the IP address or hostname of the push receiver (server) to which you want records sent. The default for this parameter is **0.0.0.0**.
3. **username**—Enter the username that the Net-Net SBC will use when it tries to send records to this push server using FTP. There is no default for this parameter.
4. **password**—Enter the password (corresponding to the username) that the Net-Net SBC will use when it tries to send records to this push server using FTP. There is no default for this parameter.

This entry works differently from other ACLI configuration parameters. For other parameters, you enter the parameter name followed by a <Space> and the value; then you press <Enter>.

- 4a. Type the parameter name **password**, and then press <Enter>.

```
ACMEPACKET(push-receiver)# password
```
- 4b. At the prompt, type the password to use when the Net-Net SBC. The display does not echo the password you enter.

```
Enter password: [enter the password]
```
- 4c. The ACLI then asks you to enter the password again. If the passwords match, then you will be returned to the user prompt to continue with configuring the push server.

```
Enter password again: [enter the password again]
ACMEPACKET(push-receiver)#

```

If the passwords do not match, then you receive an error message and must set the password again.

Error: Password mismatch - aborted.

```
ACMEPACKET(push-receiver)#

```

5. **data-store**—Enter the directory on the push receiver where you want collected data placed. There is no default for this parameter.

Starting and Stopping HDR at the Command Line

For added ease-of-use, you can stop and start record collection from the command line in Superuser Mode. You can stop and start record collection for the entire HDR process, or you can specify a group name for which you want to stop and start collection.

When you stop collection using this command, it will not restart it using this command.

To stop record collection from the command line:

1. In Superuser mode, type the ACLI **request collection** command, and the word **stop**. If you press <Enter> at this point, the Net-Net SBC will stop all record collection. If you continue with the command-line entry to specific a group-name and press <Enter>, collection for that record group only will be stopped.

```
ACMEPACKET# request collection stop volatile
```

To restart record collection from the command line:

1. In Superuser mode, type the ACLI **request collection** command, and the word **start**. If you press <Enter> at this point, the Net-Net SBC will stop all record collection. If you continue with the command-line entry to specific a group-name and press <Enter>, collection for that record group only will be stopped.

```
ACMEPACKET# request collection start volatile
```

HDR Monitoring

If there are problems with push attempts, the Net-Net SBC sends a trap. There is also a trap to clear the alarm once conditions are rectified. Refer to the *Net-Net 4000 MIB Reference Guide* for details about:

- apSysMgmtCollectorPushUnreachableTrap
- apSysMgmtCollectorPushUnreachableClearTrap

There is also collector log that provides information related to this feature's system process.

Packet Trace

Net-Net SBC Release 5.0 introduces the packet trace feature to the Net-Net SBC's capabilities. When you enable this feature, the Net-Net SBC can mirror any communication between two endpoints, or between itself and a specific endpoint. To accomplish this, the Net-Net SBC replicates the packets sent and received, and can then send them to a trace server that you designate. Using the trace server, you can display the packets on software protocol analyzer. Currently, the Net-Net SBC supports:

- One configurable trace server (on which you have installed your software protocol analyzer)
- Sixteen concurrent endpoint traces

How It Works

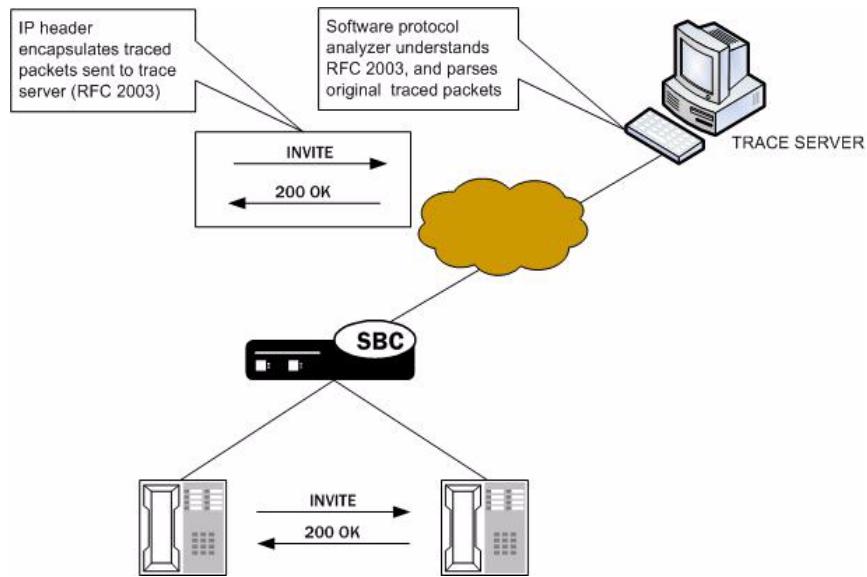
To use this feature, you configure a trace server on the Net-Net SBC so that it knows where to send the mirrored packets. Once the trace server is configured, the Net-Net SBC uses one of its internally configured IP addresses (such as one for a SIP interface or for an H.323 interface) on which to base the trace.

You start a packet trace using the ACLI Superuser command **packet-trace start**, enter with these pieces of information:

- Network interface—The name of the network interface on the Net-Net SBC from which you want to trace packets; this value can be entered either as a name alone or as a name and subport identifier value (name:subportid)
This feature is supported for front Net-Net SBC interfaces; it is not supported for rear interfaces (wancoms).
- IP address—IP address of the endpoint to and from which the Net-Net SBC will mirror calls
- Local port number—Optional parameter; Layer 4 port number on which the Net-Net SBC receives and from which it sends; if no port is specified or if it is set to 0, then all ports will be traced
- Remote port number—Optional parameter; Layer 4 port number to which the Net-Net SBC sends and from which it receives; if no port is specified or if it is set to 0, then all ports will be traced

Once the trace is initiated, the Net-Net SBC duplicates all packets sent to and from the endpoint identified by the IP address that are sent or received on the specified Net-Net SBC network interface.

The Net-Net SBC then encapsulates the original packets in accordance with RFC 2003 (IP Encapsulation within IP); it adds the requisite headers, and the payload contains the original packet trace with the Layer 2 header removed. Since software protocol analyzers understand RFC 2003, they can easily parse the original traced packets. In order to see only packet traces information in your software protocol analyzer, you can use a capture filter; for example, the Ethereal/Wireshark syntax is "ip proto 4."



It is possible that—for large frames—when the Net-Net SBC performs the steps to comply with RFC 2003 by adding the requisite header, the resulting packet might exceed Ethernet maximum transmission unit (MTU). This could result in packets being dropped by external network devices, but widespread support for jumbo frames should mitigate this possibility.

If the Net-Net SBC either receives or transmits IP fragments during a packet trace, then it will only trace the first fragment. The first fragment is likely to be a maximum-sized Ethernet frame.

The Net-Net SBC continues to conduct the packet trace and send the replicated information to the trace server until you instruct it to stop. You stop a packet trace with the ACLI **packet-trace stop** command. With this command, you can stop either an individual packet trace or all packet traces that the Net-Net SBC is currently conducting.

Packet Trace Scenarios

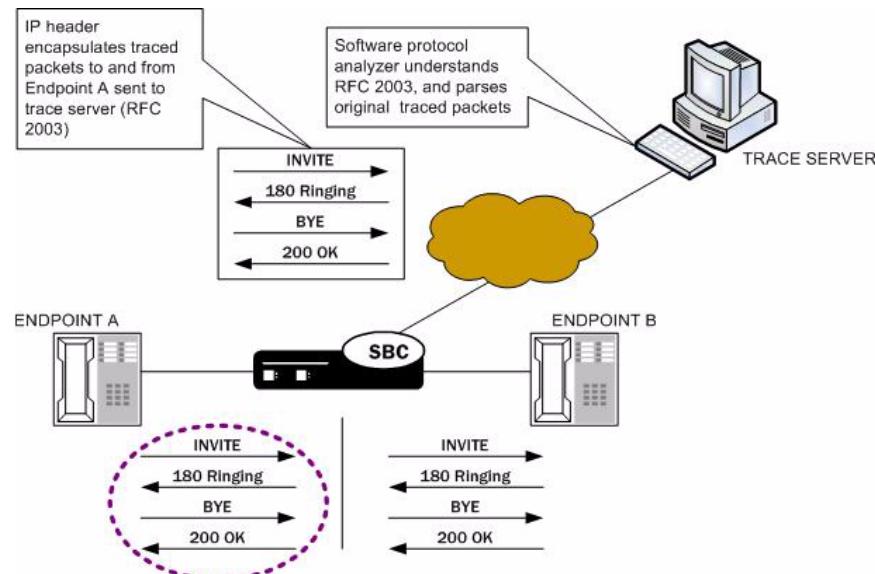
This section describes three possible ways that you might use the packet trace feature. You can examine communications sent to and from one endpoint, sent between two endpoints, or sent between ingress and/or egress Net-Net SBC interfaces to endpoints.

Packet Trace for One Endpoint

When you use the **packet-trace-state** command, the Net-Net SBC sets up packet tracing for one endpoint. The Net-Net SBC collects and replicates the packets to and from one endpoint. To enable this kind of trace, you set up one packet trace using the **packet-trace start** command.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Endpoint A>
```



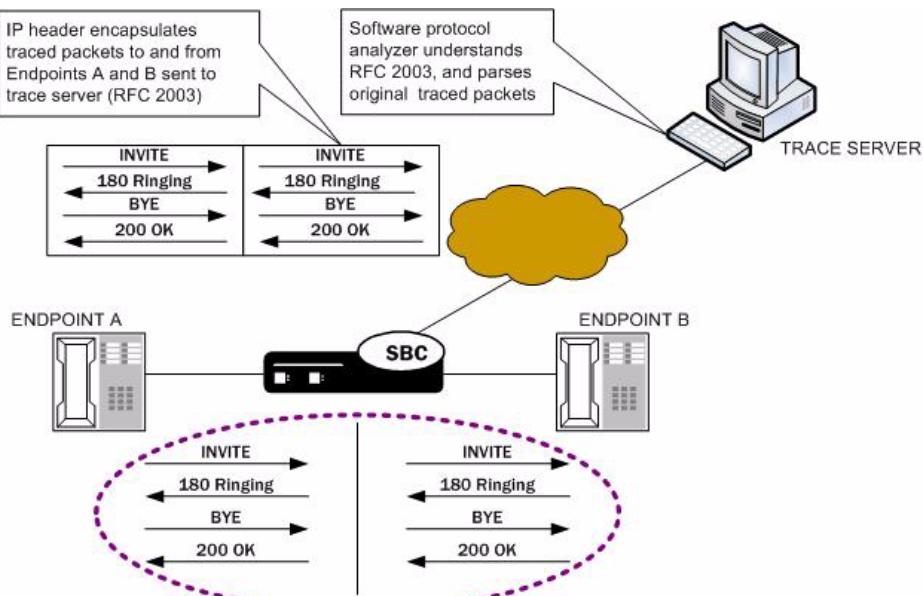
Packet Trace for Both Call Legs

If you want to trace both sides (both call legs), then you must set up individual traces for each endpoint—meaning that you would initiate two packet traces. The results of the trace will give you the communications both call legs for the communication exchanged between the endpoints you specify.

If you initiate a packet trace for both endpoints that captures both signaling and media, the signaling will be captured as usual. However, RTP will only be traced for the ingress call leg. This is because the Net-Net SBC performs NAT on the RTP, which means it cannot be captured on the egress call leg.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Endpoint A>
ACMEPACKET# packet-trace start F02 <IP address of Endpoint B>
```

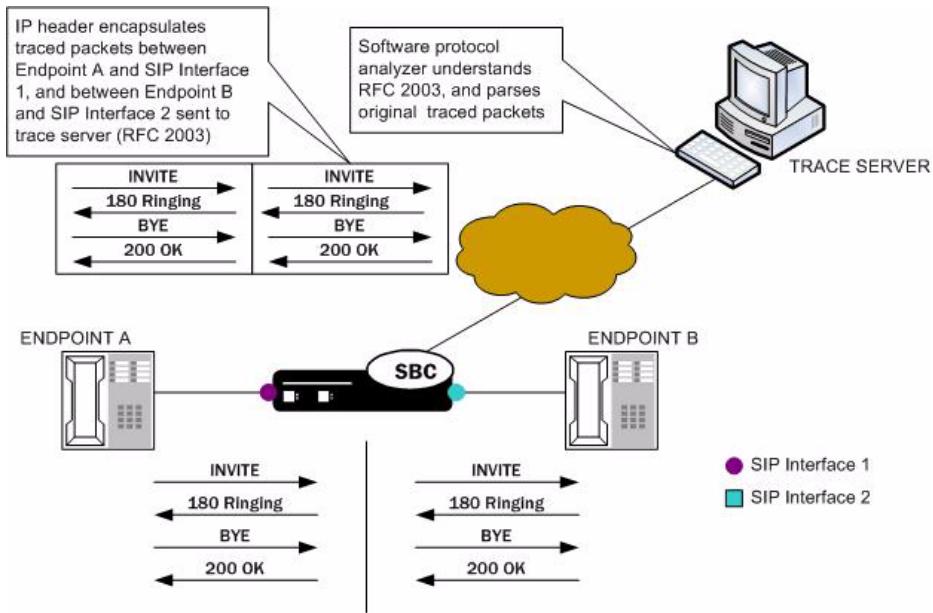


Packet Trace for a Net-Net SBC Signaling Address

You can perform a packet trace for addresses internal to the Net-Net SBC; this can be the address, for example, of a SIP or an H.323 interface. Using signaling interface addresses puts the emphasis on the Net-Net SBC rather than on the endpoints by allowing you to view traffic from specified interfaces.

The commands you carry out would take the following form:

```
ACMEPACKET# packet-trace start F01 <IP address of Net-Net SBC Interface1>
ACMEPACKET# packet-trace start F02 <IP address of Net-Net SBC Interface2>
```



ACLI Instructions and Examples

There are three steps you can take when you use the packet trace feature:

- Configuring the Net-Net SBC with the trace server information so that the Net-Net SBC knows where to send replicated data
- Setting up the capture filter “ip proto 4” in your software protocol analyzer if you only want to see the results of the Net-Net SBC packet trace(s)
- Starting a packet trace
- Stopping a packet trace

This section provides information about how to perform all three tasks.

Configuring a Trace Server

You need to configure a trace server on the Net-Net SBC; this is the device to which the Net-Net SBC sends replicated data. The Net-Net SBC supports one trace server.

To configure a trace server on your Net-Net SBC:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **system** and press <Enter>.
ACMEPACKET(configure)# **system**
ACMEPACKET(system)#[/ol>

3. Enter **capture-receiver** and press <Enter>.

```
ACMEPACKET(system)# capture-receiver
ACMEPACKET(capture receiver)#[/pre]

```
4. **state**—Type **enabled** so that you can use the trace server to which you want to send the mirrored packets for calls you are packet tracing. The default is **disabled**. The valid values are:
 - enabled | disabled
5. **address**—Enter the IP address of the trace server; there is no default.
6. **network-interface**—Enter the name and subport of the Net-Net SBC network interface from which the Net-Net SBC is to send mirrored packets. Your entry needs to take the form **name: subport**. The default is :0.
7. Save and activate your configuration.

Starting a Packet Trace

You use the start a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (**name: subport** ID combination)
- IP address to be traced; if you do not enter local and/or remote ports when you start the trace, the Net-Net SBC will trace all ports
- (Optional) Local UDP/TCP port on which the Net-Net SBC sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Net-Net SBC sends traffic, and from which it receives traffic to be traced; you cannot enter the remote port without specifying a local port

To start a packet trace with local and remote ports specified:

1. Enter the ACLI **packet-trace** command followed by a <Space>, and the word **start**. After another <Space>, type in the name and subport ID for the network interface followed by a <Space>, the IP address to be traced followed by a <Space>, the local port number followed by a <Space>, and then optionally the remote port number. Then press <Enter>.

```
ACMEPACKET# packet-trace start core:0 192.168.10.99 5060 5060
Trace started for 192.168.10.99
```

Stopping a Packet Trace

You use the stop a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (**name: subport** ID combination)
- IP address to be traced
- (Optional) Local UDP/TCP port on which the Net-Net SBC sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Net-Net SBC sends traffic, and from which it receives traffic to be traced

If the packet trace you want to stop has no entries for local and/or remote ports, then you do not have to specify them.

To stop a packet trace with local and remote ports specified:

1. Enter the ACLI **packet-trace** command followed by a <Space>, and the word **stop**. After another <Space>, type in the name and subport ID for the network

interface followed by a <Space>, the IP address to be traced followed by a <Space>, the local port number followed by a <Space>, and then optionally the remote port number. Then press <Enter>.

```
ACMEPACKET# packet-trace stop core: 0 192.168.10.99 5060 5060
```

To stop all packet traces on the Net-Net SBC:

1. Enter the ACCLI **packet-trace** command followed by a <Space>, and the word **stop**. After another <Space>, type the word **all** and press <Enter>.

```
ACMEPACKET# packet-trace stop all
```

RAMdrive Log Cleaner

The RAMdrive log cleaner allows the Net-Net SBC to remove log files proactively and thereby avoid situations where running low on RAMdrive space is a danger. Because even a small amount of logging can consume a considerable space, you might want to enable the RAMdrive log cleaner.

How It Works

The RAMdrive cleaner periodically checks the remaining free space in the RAMdrive and, depending on the configured threshold, performs a full check on the /ramdrv/logs directory. During the full check, the RAMdrive cleaner determines the total space logs files are using and deletes log files that exceed the configured maximum lifetime. In addition, if the cleaner finds that the maximum log space has been exceeded or the minimum free space is not sufficient, it deletes older log files until the thresholds are met.

Not all log files, however, are as active as others. This condition affects which log files the log cleaner deletes to create more space in RAMdrive. More active log files rotate through the system more rapidly. So, if the log cleaner were to delete the oldest of these active files, it might not delete less active logs files that could be older than the active ones. The log cleaner thus deletes files that are truly older, be they active or inactive.

Applicable Settings

In the system configuration, you establish a group of settings in the options parameter that control the log cleaner's behavior:

- **ramdrv-log-min-free**—Minimum percent of free space required when rotating log files.
When the amount of free space on the RAMdrive falls below this value, the log cleaner deletes the oldest copy of the log file. The log cleaner also uses this setting when performing period cleaning.
- **ramdrv-log-max-usage**—Maximum percent of the RAMdrive the log files can use.
The log cleaner removes old log files to maintain this threshold.
- **ramdrv-log-min-check**—Minimum percent of free space on the RAMdrive that triggers the log cleaner to perform a full check of log files.
- **ramdrv-min-log-check**—Minimum time (in seconds) between log cleaner checks.
- **ramdrv-max-log-check**—Maximum time (in seconds) between log cleaner checks. This value must be greater than or equal to the **ramdrv-min-log-check**.
- **ramdrv-log-lifetime**—Maximum lifetime (in days) for log files. You give logs unlimited lifetime by entering a value of 0.

Clean-Up Procedure

The log cleaner checks the amount of space remaining in the RAMdrive and performs a full check of the logs directory when:

- Free space is less than the minimum percent of the RAMdrive that triggers a full check of log files
- The amount of free space has changed by more than 5% of the RAMdrive capacity since the last full check
- A full check of the logs directory has not been performed in the last hour

When it checks the logs directory, the log cleaner inventories the collected log files. It identifies each files as one of these types:

- Process log—Files beginning with `log`.
- Internal trace file—A `<task>.log` file
- Protocol trace file—Call trace including `si pmsg.log`, `dns.log`, `si pddns.log`, and `alg.log`
- CDR file—File beginning with `cdr`

Next, the log cleaner determines the age of the log files using the number of seconds since the log files were created. Then it orders the files from oldest to newest. The age adjusts such that it always increases as the log file sequence number (a suffix added by file rotation) increases. The log cleaner applies an additional weighting factor to produce a weighted age that favors the preservation of protocol traces files over internal trace files, and internal trace files over process log files. The base log file and CDR files are excluded from the age list and so will not be deleted; the accounting configuration controls CDR file aging.

With the age list constructed, the log cleaner examines the list from highest weighted age to lowest. If the actual file age exceeds the RAMdrive maximum log lifetime, the log cleaner deletes it. Otherwise, the log cleaner deletes files until the maximum percent of RAMdrive that logs can use is no longer exceeded and until the minimum percent of free space required when rotating logs is available.

Clean-Up Frequency

The minimum free space that triggers a full check of log files and the maximum time between log file checks control how often the log cleaner performs the clean-up procedure. When it completes the procedure, the log cleaner determines the time interval until the next required clean-up based on the RAMdrive's state.

If a clean-up results in the deletion of one or more log files or if certain thresholds are exceeded, frequency is based on the minimum time between log cleaner checks. Otherwise, the system gradually increases the interval up to the maximum time between log cleaner checks. The system increases the interval by one-quarter of the difference between the minimum and maximum interval, but not greater than one-half the minimum interval or smaller than 10 seconds. For example, using the default values, the interval would be increased by 30 seconds.

ACLI Instructions and Examples

You configure the log cleaner's operating parameters and thresholds in the system configuration. Note that none of these settings is RTC-supported, so you must reboot your Net-Net SBC in order for them to take effect. If you are using this feature on an HA node, however, you can add this feature without impact to service by activating the configuration, rebooting the standby, switching over to make the newly booted standby active, and then rebooting the newly standby system.

Unlike other values for **options** parameters, the Net-Net SBC validates these setting when entered using the ACLI. If any single value is invalid, they all revert to their default values.

To configure the RAMdrive log cleaner:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

2. Type **system** and press <Enter>.

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

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

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

4. **options**—Set the options parameter by typing options, a <Space>, <option name>=X (where X is the value you want to use) with a “plus” sign in front of it. Then press <Enter>.

Remember that if any of your settings are invalid, the Net-Net SBC changes the entire group of these options back to their default settings.

Option Name	Description
ramdrv-log-min-free	Minimum percent of free space required when rotating log files. When the amount of free space on the RAMdrive falls below this value, the log cleaner deletes the oldest copy of the log file. The log cleaner also uses this setting when performing period cleaning. Default=40; Minimum=15; Maximum=75
ramdrv-log-max-usage	Maximum percent of the RAMdrive the log files can use. The log cleaner removes old log files to maintain this threshold. Default=40; Minimum=15; Maximum=75
ramdrv-log-min-check	Minimum percent of free space on the RAMdrive that triggers the log cleaner to perform a full check of log files. Default=50; Minimum=25; Maximum=75
ramdrv-min-log-check	Maximum time (in seconds) between log cleaner checks. This value must be greater than or equal to the ramdrv-min-log-check. Default=180; Minimum=40; Maximum=1800
ramdrv--log-lifetime	Maximum lifetime (in days) for log files. You give logs unlimited lifetime by entering a value of 0. Default=30; Minimum=2; Maximum=9999

```
ACMEPACKET(system-config)# options +ramdrv-log-min-free=50
ACMEPACKET(system-config)# options +ramdrv-log-max-usage=50
ACMEPACKET(system-config)# options +ramdrv-log-min-check=35
ACMEPACKET(system-config)# options +ramdrv-min-log-check=120
```

```
ACMEPACKET(system-config)# options +ramdrv-max-log-free=1500
ACMEPACKET(system-config)# options +ramdrv-log-life=7
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Reboot your Net-Net SBC.

Alarm Synchronization

Two trap tables in the `ap-smgmt.mib` record trap information for any condition on the Net-Net SBC that triggers an alarm condition. 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 that 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 multiples of the same trap type that have occurred within a second of one another and have different information. Each table entry contains the following:
 - Trap identifier
 - System time (synchronized with an NTP server)
 - `sysUpTime`
 - Instance number
 - Other trap information for this trap identifier
- `apSysMgmtTrapInformationTable`—You can poll this table to obtain further details about the traps recorded in the `apSysMgmtTrapTable` table. The following information appears:
 - Data index
 - Data type
 - Data length
 - The data itself (in octets)

Trap tables do not record information about alarm severity.

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

Caveats

Note that the Net-Net SBC does not replicate alarm synchronization table data across HA nodes. That is, each Net-Net SBC in an HA node maintains its own tables.

ACLI Instructions and Examples

You turn on alarm synchronization in the system configuration.

To use alarm synchronization:

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

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

```
ACMEPACKET(config)# 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 Configuration

The Net-Net SBC offers support for RADIUS, an accounting, authentication, and authorization (AAA) system. In general, RADIUS servers are responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver service to the user.

You can configure your Net-Net SBC to send call accounting information to one or more RADIUS servers. This information can help you to see usage and QoS metrics, monitor traffic, and even troubleshoot your system. For more information about QoS, refer to the *Admission Control and QoS* chapter of this guide.

For information about how to configure the Net-Net SBC for RADIUS accounting use, refer to the *Net-Net 4000 Accounting Guide*. This guide contains all RADIUS information, as well as information about:

- Accounting for SIP and H.323
- Local CDR storage on the Net-Net SBC, including CSV file format settings
- The ability to send CDRs via FTP to a RADIUS sever (the FTP push feature)
- Per-realm accounting control
- Configurable intermediate period
- RADIUS CDR redundancy
- RADIUS CDR content control

SIP over SCTP

In releases prior to Release S-C6.1.0, the Net-Net SBC supports UDP and TCP as transport protocols for SIP signaling. Release S-C6.1.0 introduces support for Stream Control Transport Protocol (SCTP). Young in relation to UDP and TCP, SCTP seeks to address some of the shortcomings of the other two transport protocols—most notably by supporting multi-homing and multi-streaming.

For a full description of SCTP, refer to RFC 2960 Stream Control Transmission Protocol.

SCTP Concepts

This section defines some terms commonly found in descriptions of SCTP. You might find them useful.

SCTP Term	Definition
SCTP association	Refers to a communication relationship (or logical connection) between SCTP endpoints. SCTP uses a four-way handshake to establish the association between endpoints. This handshake is similar to three-way handshake TCP uses.
Multi-homing	Refers to instances when multiple IP addresses are assigned to a host on the network. Typically, this arrangement entails a host that has multiple network interface cards. (To be supported in future Net-Net OS releases)
Multi-streaming	Refers to the ability to partition data within an association into multiple logical communication channels. Each logical channel—or stream—has the property of independent sequenced delivery. This means that data loss on one stream has no impact on delivery on other streams. (Release S-C6.1.0 supports two incoming streams and two outgoing streams.)
SCTP endpoint	Refers to a logical sender and/or receiver of SCTP packets.

SCTP Overview and Comparisons

A connection-oriented protocol, SCTP uses a four-way handshake to create a connection between two peer entities. This handshake is similar to the three-way handshake TCP uses. SCTP uses the following messages to establish an association:

1. INIT—Message the client endpoint sends to initiate an association with a peer endpoint
2. INIT-ACK—Message acknowledging the INIT; includes a cookie
3. COOKIE-ECHO—Message that is an echo of the cookie received in the INIT-ACK
4. COOKIE-ACK—Message acknowledging the COOKIE-ECHO

On successful completion of this handshake, the association is established.

How Is SCTP Different from TCP?

While SCTP has many advantages over TCP, the most advertised are multi-streaming and multi-homing.

Multi-streaming allows SCTP to overcome the “head of line” blocking issue that can happen in TCP. In TCP, the loss of a data segment can prevent the delivery of subsequent segments until the lost segment is recovered (this is head of line blocking). SCTP circumvents this problem by supporting multiple associated streams. Data within an association is divided into multiple streams, providing independent and ordered delivery of the data. And so data loss from one stream is prevented from interfering with the data delivery on other streams.

Multi-homing means that an SCTP endpoint can support multiple IP addresses on the same host. In SCTP, application layer software chooses one of the IP address as the primary address. The endpoint should always use the primary address by default,

unless specified to do otherwise by the application layer software. If an SCTP endpoint has data to transmit but its primary address is unavailable, the endpoint attempts to transmit its data using one of the alternative IP addresses. This creates a redundancy mechanism transparent to the application layer software.

One minor but notable advantage of SCTP or TCP is the former's support for message-oriented communication. TCP uses stream-oriented communication that requires the application layer to ascertain message boundaries. SCTP's message-based communication clearly identifies the beginning and end of data messages.

How Is SCTP Different from UDP?

Both SCTP and TCP are connection-oriented protocols that share some common advantages over UDP: reliable data transfer, congestion control, transport layer fragmentation. Multi-streaming and multi-homing are also SCTP's advantages over UDP.

ACLI Instructions and Examples

The Net-Net SBC uses the SIP Via header in the to determine if SCTP should be used as the transport protocol. Minor changes to the ACLI have been made to support SCTP's use.

Configuration and Parameter	New Value
sip-interface>sip-port	SCTP
session-agent>transport-method	StaticSCTP—Static connections are persistent and will automatically attempt re-connection if a failure occurs.
session-agent>reuse-connections	SCTP—Allows for reuse of SCTP connections.

Setting the SCTP Delivery Mode

In addition, you can also set an SCTP delivery mode to:

- Ordered—Meaning that the endpoint must deliver data according to the order of their stream sequence number
- Unordered—Meaning that the endpoint can deliver data within regard for their stream sequence number

You set this preference in the network parameters configuration.

To set the SCTP delivery mode:

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

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

```
2. Type **system** and press <Enter>.

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

```
3. Type **network-parameters** and press <Enter>.

```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#

```
4. **sctp-send-mode**—Leave this parameter set to its default (unordered) so data delivery can occur without regard to stream sequence numbering. If data

delivery must follow stream sequence number, change this parameter to **ordered**.

5. Save and activate your configuration.

About Your Net-Net 3800/4500 and IPv6

IPv6 support has been added to the Net-Net 3800 and Net-Net 4500. Ideally, IPv6 support would be a simple matter of configuring IP addresses of the version type you want in the configurations where you want them. While this is the case for some configuration areas, in others you will need to take care with—for example—the format of your IPv6 address entries or where parameters must be configured with IP addresses of the same version type.

This section explains the changes to the ACLI of which you need to be aware as you start to use IPv6 on your Net-Net 3800 or 4500. Note that this first-available implementation of IPv6 is expected to expand in the future; not all configurations and their parameters are available for IPv6 use.

RTN 1752

Licensing

IPv6 is a licensed feature on the Net-Net 3800 and Net-Net 4500. If you want to add this license to a system, then contact your Acme Packet sales engineering for information related to the license. Once you have the license information, refer to the Getting Started chapter of the Net-Net 4000 ACLI Configuration Guide of instructions about how to add a license.

You do not need to take action if you are working with a new system with which the IPv6 license was purchased.

Globally Enabling IPv6

To use IPv6 on your Net-Net 3800 or 4500, you need to set the **ipv6-support** parameter in the **system-config**, and then you must reboot your system.

Remember that if you reboot your Net-Net SBC from a Telnet session, you lose IP access and therefore your connection.

To enable your system for IPv6 support:

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

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

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

```
3. Type **system-config** and press <Enter>.

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

```
4. **ipv6-support**—Set this parameter to **enabled** if you want to use IPv6 on your system. Otherwise, you can leave this parameter set to **disabled** (default).
5. Save your work.

6. Type **exit** at the system prompt until reach the main Superuser level.
 7. Reboot your Net-Net 3800/4500 for changes to take effect.
- The ACLI **reboot** and **reboot force** commands initiate a reboot. With the **reboot** command, you must confirm that you want to reboot. With the **reboot force** command, you do not have make this confirmation.
- ```
ACMEPACKET# reboot force
```
8. The Net-Net SBC completes the full booting sequence.

## Updated ACLI Help Text

---

As you complete configuration work and perform monitoring tasks on your system, you might note that there have been changes to the help text to reflect the addition of IPv6 support. These changes are minor, but nonetheless reflect feature support.

In the ACLI that supports only IPv4, there are many references to that version as the accepted value for a configuration parameter or other IPv4-specific languages. For IPv6 support, these references have been edited. For example, rather than providing help that refers specifically to IPv4 addresses when explaining what values are accepted in an ACLI configuration parameter, you will now see an <ipAddr> note.

## IPv6 Address Configuration

---

This section calls out the configurations and parameters for which you can enter IPv6 addresses. In this first IPv6 implementation, the complete range of system configurations and their parameters are available for IPv6 use.

The Net-Net SBC follows RFC 3513 its definition of IPv6 address representations. Quoting from that RFC, these are the two forms supported:

- The preferred form is x:x:x:x:x:x, where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address. Examples:

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

1080:0:0:0:8:800:200C:417A

Note that it is not necessary to write the leading zeros in an individual field, but there must be at least one numeral in every field (except for the case described in 2.).

- Due to some methods of allocating certain styles of IPv6 addresses, it will be common for addresses to contain long strings of zero bits. In order to make writing addresses containing zero bits easier a special syntax is available to compress the zeros. The use of ":" indicates one or more groups of 16 bits of zeros. The ":" can only appear once in an address. The ":" can also be used to compress leading or trailing zeros in an address. For example, the following addresses: 1080:0:0:0:8:800:200C:417A a unicast address FF01:0:0:0:0:0:101 a multicast address

0:0:0:0:0:0:1           the loopback address

0:0:0:0:0:0:0           the unspecified addresses

may be represented as:

1080::8:800:200C:417A    a unicast address

FF01::101                a multicast address

::1                      the loopback address

::                        the unspecified addresses

**Access Control**

These are the IPv6-enabled parameters in the **access-control** configuration.

| Parameter           | Entry Format                                     |
|---------------------|--------------------------------------------------|
| source-address      | <ip-address>[/<num-bits>][:<port>[/<port-bits>]] |
| destination-address | <ip-address>[/<num-bits>][:<port>[/<port-bits>]] |

**Host Route**

These are the IPv6-enabled parameters in the **host-route** configuration.

| Parameter    | Entry Format    |
|--------------|-----------------|
| dest-network | <ipv4>   <ipv6> |
| netmask      | <ipv4>   <ipv6> |
| gateway      | <ipv4>   <ipv6> |

**Local Policy**

These are the IPv6-enabled parameters in the **local-policy** configuration.

| Parameter    | Entry Format                                                    |
|--------------|-----------------------------------------------------------------|
| from-address | <ipv4>   <ipv6>   POTS Number, E.164 Number, hostname, wildcard |
| to-address   | <ipv4>   <ipv6>   POTS Number, E.164 Number, hostname, wildcard |

**Network Interface**

These are the IPv6-enabled parameters in the **network-interface** configuration.

| Parameter        | Entry Format               |
|------------------|----------------------------|
| hostname         | <ipv4>   <ipv6>   hostname |
| ip-address       | <ipv4>   <ipv6>            |
| pri-utility-addr | <ipv4>   <ipv6>            |
| sec-utility-addr | <ipv4>   <ipv6>            |
| netmask          | <ipv4>   <ipv6>            |
| gateway          | <ipv4>   <ipv6>            |
| sec-gateway      | <ipv4>   <ipv6>            |
| dns-ip-primary   | <ipv4>   <ipv6>            |

| Parameter      | Entry Format    |
|----------------|-----------------|
| dns-ip-backup1 | <ipv4>   <ipv6> |
| dns-ip-backup2 | <ipv4>   <ipv6> |
| add-hip-ip     | <ipv4>   <ipv6> |
| remove-hip-ip  | <ipv4>   <ipv6> |
| add-icmp-ip    | <ipv4>   <ipv6> |
| remove-icmp-ip | <ipv4>   <ipv6> |

## Realm Configuration

These are the IPv6-enabled parameters in the **realm-config**.

| Parameter   | Entry Format             |
|-------------|--------------------------|
| addr-prefix | [<ipv4>   <ipv6>]/prefix |

## Session Agent

These are the IPv6-enabled parameters in the **session-agent** configuration.

| Parameter  | Entry Format    |
|------------|-----------------|
| hostname   | <ipv4>   <ipv6> |
| ip-address | <ipv4>   <ipv6> |

## SIP Configuration

These are the IPv6-enabled parameters in the **session-config**.

| Parameter      | Entry Format                   |
|----------------|--------------------------------|
| registrar-host | <ipv4>   <ipv6>   hostname   * |

## SIP Interface>SIP Ports

These are the IPv6-enabled parameters in the **sip-interface>sip-ports** configuration.

| Parameter | Entry Format    |
|-----------|-----------------|
| address   | <ipv4>   <ipv6> |

## Steering Pool

These are the IPv6-enabled parameters in the **steering-pool** configuration.

| Parameter  | Entry Format    |
|------------|-----------------|
| ip-address | <ipv4>   <ipv6> |

## System Configuration

These are the IPv6-enabled parameters in the **system-config**.

| Parameter          | Entry Format |
|--------------------|--------------|
| default-v6-gateway | <ipv6>       |

## IPv6 Default Gateway

In the system configuration, you configure a default gateway—a parameter that now has its own IPv6 equivalent.

### To configure an IPv6 default gateway:

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

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

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

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)#
```
4. **default-v6-gateway**—Set the IPv6 default gateway for this Net-Net SBC. This is the IPv6 egress gateway for traffic without an explicit destination. The application of your Net-Net SBC determines the configuration of this parameter.
5. Save your work.

## Network Interfaces and IPv6

You set many IP addresses in the network interface, one of which is the specific IP address for that network interface and others that are related to different types of management traffic. This section outlines rules you must follow for these entries.

- For the **network-interface ip-address** parameter, you can set a single IP address. When you are working with an IPv6-enabled system, however, note that all other addresses related to that network-interface IP address must be of the same version.
- Heterogeneous address family configuration is prevented for the **dns-ip-primary**, **dns-ip-backup1**, and **dns-ip-backup2** parameters.
- For HIP addresses (**add-hip-ip**), you can use either IPv4 or IPv6 entries.

- For ICMP addresses (`add-icmp-ip`), you can use either IPv4 or IPv6 entries.
- For Telnet (`add-telnet-ip`), FTP (`add-ftp-ip`), and SNMP (`add-snmp-ip`), you are not allowed to use IPv6; your entries MUST use IPv4.

## Access Control List Support

---

The Net-Net SBC supports IPv6 for access control lists in two ways:

- For static access control lists that you configure in the `access-control` configuration, your entries can follow IPv6 form. Further, this configuration supports a prefix that enables wildcarding the source IP address.
- Dynamic ACLs are also supported; the Net-Net SBC will create ACLs for offending IPv6 endpoints.

## Data Entry

When you set the `source-address` and `destination-address` parameters in the `access-control` configuration, you will use a slightly different format for IPv6 than for IPv4.

For the `source-address`, your IPv4 entry takes the following format: `<i p-address>[/<num-bits>[:<port>[/<port-bits>]]`. And for the `destination-address`, your IPv4 entry takes this format: `<i p-address>[:<port>[/<port-bits>]]`.

Since the colon (:) in the IPv4 format leads to ambiguity in IPv6, your IPv6 entries for these settings must have the address encased in brackets ([]):

`[7777::11]/64:5000/14`.

In addition, IPv6 entries are allowed up to 128 bits for their prefix lengths.

The following is an example access control configuration set up with IPv6 addresses.

```
ACMEPACKET(access-control)# done
access-control
 realm-id net7777
 description
 source-address 7777::11/64:5060/8
 destination-address 8888::11:5060/8
 application-protocol SIP
 transport-protocol ALL
 access deny
 average-rate-limit 0
 trust-level none
 minimum-reserved-bandwidth 0
 invalid-signal-threshold 10
 maximum-signal-threshold 0
 untrusted-signal-threshold 0
 deny-period 30
```

## DNS Support

---

The Net-Net SBC supports the DNS resolution of IPv6 addresses; in other words, it can request the AAAA record type (per RFC 1886) in DNS requests. In addition, the Net-Net SBC can make DNS requests over IPv6 transport so that it can operate in networks that host IPv6 DNS servers.

For mixed IPv4-IPv6 networks, the Net-Net SBC follows these rules:

- If the realm associated with the name resolution is an IPv6 realm, the Net-Net SBC will send the query out using the AAAA record type.

- If the realm associated with the name resolution is an IPv4 realm, the Net-Net SBC will send the query out using the A record type.

In addition, heterogeneous address family configuration is prevented for the `dns-ip-primary`, `dns-ip-backup1`, and `dns-ip-backup2` parameters.

## Homogeneous Realms

---

IPv6 is supported for realms and for nested realms, as long as the parent chain remains within the same address family. If you try to configure realms with mixed IPv4-IPv6 addressing, your system will issue an error message when you try to save your configuration. This check saves you time because you do not have to wait to run a configuration verification (using the ACLI `verify-config` command) to find possible errors.

### Parent-Child Network Interface Mismatch

Your system will issue the following error message if parent-child realms are on different network interfaces that belong to different address families:

```
ERROR: realm-config [child] and parent [net8888] are on network
 interfaces that belong to different address families
```

### Address Prefix-Network Interface Mismatch

If the address family and the address-prefix you configure for the realm does not match the address family of its network interface, your system will issue the following error message:

```
ERROR: realm-config [child] address prefix and network interface
 [1:1:0] belong to different address families
```

## RADIUS Support for IPv6

---

The Net-Net SBC's RADIUS support now includes:

- RADIUS CDR generation for SIPv6-SIPv6 and SIPv6-SIPv4 calls
- IPv6-based addresses in RADIUS CDR attributes

This means that for the CDR attributes in existence prior to the introduction of IPv6 to the Net-Net 3800/4500 are mapped to the type `i paddr`, which indicates four-byte field. The sixteen-byte requirement for IPv6 addresses is now supported, and there are a parallel set of attributes with the type `i pv6addr`. Attributes 155-170 are reserved for the IPv6 addresses.

NAS addresses use the number 95 to specify the NAS-IPV6-Address attribute. And local CDRs now contain IPv6 addresses.

### Supporting RADIUS VSAs

The following VSAs have been added to the Acme Packet RADIUS dictionary to support IPv6.

|                                                |     |                        |      |
|------------------------------------------------|-----|------------------------|------|
| <code>Acme-FIow-In-Src-IPv6_Addr_FS1_F</code>  | 155 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-In-Dst-IPv6_Addr_FS1_F</code>  | 156 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-Out-Src-IPv6_Addr_FS1_F</code> | 157 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-Out-Dst-IPv6_Addr_FS1_F</code> | 158 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-In-Src-IPv6_Addr_FS1_R</code>  | 159 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-In-Dst-IPv6_Addr_FS1_R</code>  | 160 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-Out-Src-IPv6_Addr_FS1_R</code> | 161 | <code>i pv6addr</code> | Acme |
| <code>Acme-FIow-Out-Dst-IPv6_Addr_FS1_R</code> | 162 | <code>i pv6addr</code> | Acme |

|                                   |     |           |      |
|-----------------------------------|-----|-----------|------|
| Acme-FIow-In-Src-IPv6_Addr_FS2_F  | 163 | i_pv6addr | Acme |
| Acme-FIow-In-Dst-IPv6_Addr_FS2_F  | 164 | i_pv6addr | Acme |
| Acme-FIow-Out-Src-IPv6_Addr_FS2_F | 165 | i_pv6addr | Acme |
| Acme-FIow-Out-Dst-IPv6_Addr_FS2_F | 166 | i_pv6addr | Acme |
| Acme-FIow-In-Src-IPv6_Addr_FS2_R  | 167 | i_pv6addr | Acme |
| Acme-FIow-In-Dst-IPv6_Addr_FS2_R  | 168 | i_pv6addr | Acme |
| Acme-FIow-Out-Src-IPv6_Addr_FS2_R | 169 | i_pv6addr | Acme |
| Acme-FIow-Out-Dst-IPv6_Addr_FS2_R | 170 | i_pv6addr | Acme |



## 4

# Realms and Nested Realms

## Introduction

---

This chapter explains how to configure realms and nested realms, and specialized media-related features.

A realm is a logical definition of a network or groups of networks made up in part by devices that provide real-time communication sessions comprised of signaling messages and possibly media flows. These network devices might be call agents, softswitches, SIP proxies, H.323 gatekeepers, IP PBXs, etc., that are statically defined by IPv4 addresses. These network devices might also be IPv4 endpoints: SIP phones, IADs, MAs, media gateways, etc., that are defined by an IPv4 address prefix.

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

This chapter also explains how to configure media ports (steering pools). A steering pool exists within a realm and contains a range of ports that have a common address (for example, a target IPv4 address). The range of ports contained in the steering pool are used to steer media flows from one realm, through the Net-Net SBC, 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 SBC and what kinds of resources and special functions apply to those routes. Realms are used as a basis for determining ingress and egress associations to network interfaces, which can reside in different VPNs. The ingress realm is determined by the signaling interface on which traffic arrives. The egress realm is determined by the following:

- Routing policy—Where the egress realm is determined in the session agent configuration or external address of a SIP-NAT
- Realm-bridging—As applied in the SIP-NAT configuration and H.323 stack configurations
- Third-party routing/redirect (i.e., SIP redirect or H.323 LCF)

Realms also provide configuration support for denial of service (DoS)/access control list (ACL) functionality. For more information about the Net-Net SBC'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 SBC.

## About Realms and Network Interfaces

All realms reference network interfaces on the Net-Net SBC. 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 SBC. 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 SBC'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 SBC'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 SBC Functions

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

- SIP configuration: home realm identifier, egress realm identifier
- SIP-NAT configuration: realm identifier
- H.323 stack configuration: realm identifier
- MGCP configuration: private realm, public realm
- Session agent configuration: realm identifier
- Media manager: home realm identifier
- Steering ports: realm identifier
- Static flow: in realm identifier, out realm identifier

# 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 SBC 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 SBC 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 ACCLI:

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(real-m-config)#[
```

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

## Identity and IP Address Prefix

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

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

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

1. **identifier**—Enter the name of the realm. This parameter uniquely identifies the realm. You will use this parameter in other configurations when asked for a realm identifier value.
2. **addr-prefix**—Enter the IPv4 address and subnet mask combination to set the criteria the Net-Net SBC 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 IPv4 address comes first and is separated by a slash (/) from the subnet mask value. For example, 172. 16. 0. 0/24.

The default for this parameter is 0.0.0.0. When you leave this parameter set to the default, all addresses match.

**Realm Interfaces**

The realm points to one or more network interfaces on the Net-Net SBC. 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.

**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, f10:0.

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

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

```
ACMEPACKET(real m-confi g)# network-i nterfaces (fe1:0 fe2: 1)
```

**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 SBC'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 SBC. Media travels through the Net-Net SBC rather than straight between the endpoints.
- Media can be directed through the Net-Net SBC between endpoints that are in different realms, but share the same subnet.
- For SIP only, media can be released between multiple Net-Net SBCs.

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.

#### To configure realm service profile:

1. **max-bandwidth**—Enter the total bandwidth budget in kilobits per second for all flows to/from the realm defined in this element. The default is **0** which allows for unlimited bandwidth. The valid range is:
  - Minimum—0
  - Maximum— $2^{32}-1$
2. **mm-in-realm**—Enable this parameter to treat media within this realm on this Net-Net SBC. The default is **disabled**. Valid values are:
  - enabled | disabled
3. **mm-in-network**—Enable this parameter to treat media within realms that have the same subnet mask on this Net-Net SBC. The default is **enabled**. Valid values are:
  - enabled | disabled
4. **msm-release**—Enable or disable the inclusion of multi-system (multiple Net-Net SBCs) media release information in the SIP signaling request sent into the realm identified by this realm-config element. If this field is set to enabled, another Net-Net SBC is allowed to decode the encoded SIP signaling request message data sent from a SIP endpoint to another SIP endpoint in the same network to restore the original SDP and subsequently allow the media to flow directly between those two SIP endpoints in the same network serviced by multiple Net-Net SBCs. If this field is disabled, the media and signaling will pass through both Net-Net SBCs. Remember that for this feature to work, you must also set the options parameter in the SIP configuration accordingly. The default is **disabled**. Valid values are:
  - enabled | disabled

#### **QoS Measurement**

Refer to this guide's *Admission Control and QoS* chapter for more information about enabling QoS measurements on your Net-Net SBC. 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 SBC, then you can leave this parameter set to **disabled** (default).

#### **QoS Marking**

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

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

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

- One or more audio media types for SIP and/or H.323
- One or more video types for SIP and/or H.323
- Both audio and video media types for SIP and/or H.323

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

1. **media-policy**—Enter the name (unique identifier) of the media policy you want to apply in the realm. When the Net-Net SBC first sets up a SIP or H.323 media session, it identifies the egress realm of each flow and then determines the media-policy element to apply to the flow. This parameter must correspond to a valid name entry in a media policy element. If you leave this parameter empty, then QoS marking for media will not be performed for this realm.

## **Address Translation Profiles**

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

## **DNS Servers**

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

This feature is available for SIP and MGCP only.

### **To configure realm-specific DNS in the ACLI:**

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 SBC. If you are not using this functionality, you can leave the parameters at their default values: **average-rate-limit**, **peak-rate-limit**, **maximum-burst-size**, **access-control-trust-level**, **invalid-signal-threshold**, and **maximum-signal-threshold**.

## **Enabling RTP-RTCP UDP Checksum Generation**

You can configure your Net-Net SBC to generate a UDP checksum for RTP/ RTCP packets on a per-realm basis. This feature is useful in cases where devices performing network address translation (NATs) do not pass through packets with a zero checksum from the public Internet. These packets do not make it through the NAT even if they have the correct to and from IP address and UDP port information. When you enable this feature, the Net-Net SBC calculates a checksum for these packets and thereby enables them to traverse a NAT successfully.

If you do not enable this feature, then the Net-Net SBC will not generate a checksum for RTP or RTCP packets if their originator did not include one. If a checksum is already present when the traffic arrives at the Net-Net 4000, the system will relay it.

You enable this feature on the outbound realm.

## **Aggregate Session Constraints Per Realm**

You can set session constraints for the Net-Net SBC's global SIP configuration, specified session agents, and specified SIP interfaces. This forces users who have a large group of remote agents to create a large number of session agents and SIP interfaces.

With this feature implemented, however, you can group remote agents into one or more realms on which to apply session constraints.

**To enable sessions constraints on a per realm basis:**

1. **constraint-name**—Enter the name of the constraint you want to use for this realm. You set up in the session-constraints configuration; for more information about them, refer to the [Aggregate Session Constraints for SIP \(834\)](#) section in this guide's *Admission Control and Quality of Service Reporting* chapter.

**ACLI Instructions and Examples****Admission Control Configuration****Reserved Parameters****To enable UDP checksum generation for a realm:**

1. **generate-udp-checksum**—Enable this parameter to generate a UDP checksum for this outbound realm. The default is **disabled**. Valid values are:
  - enabled | disabled

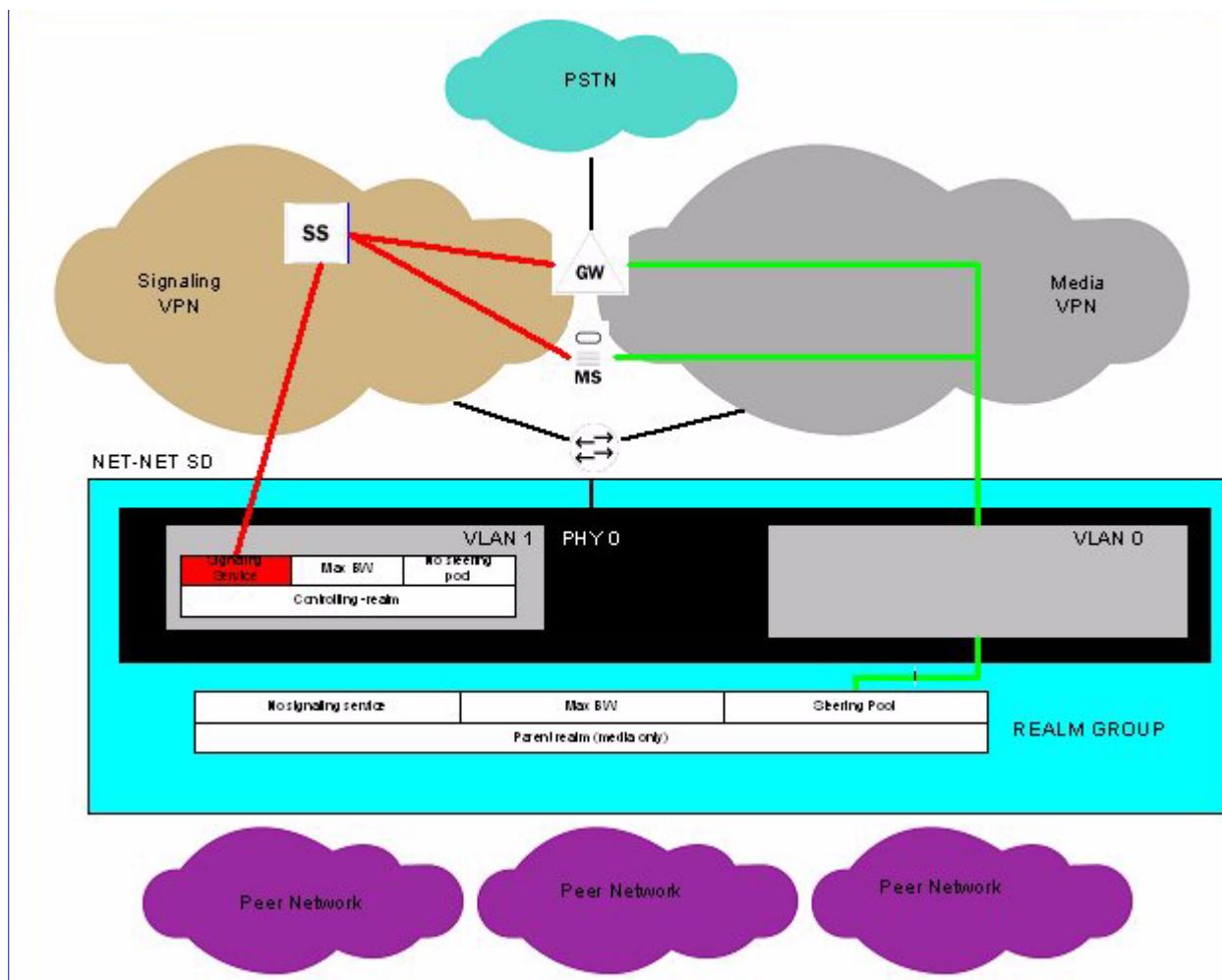
You can set admission control based on bandwidth for each realm by setting the **max-bandwidth** parameter for the realm configuration. Details about admission control are covered in this guide's *Admission Control and QoS* chapter.

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

## Nested Realms

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

The following figure shows the network architecture.



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

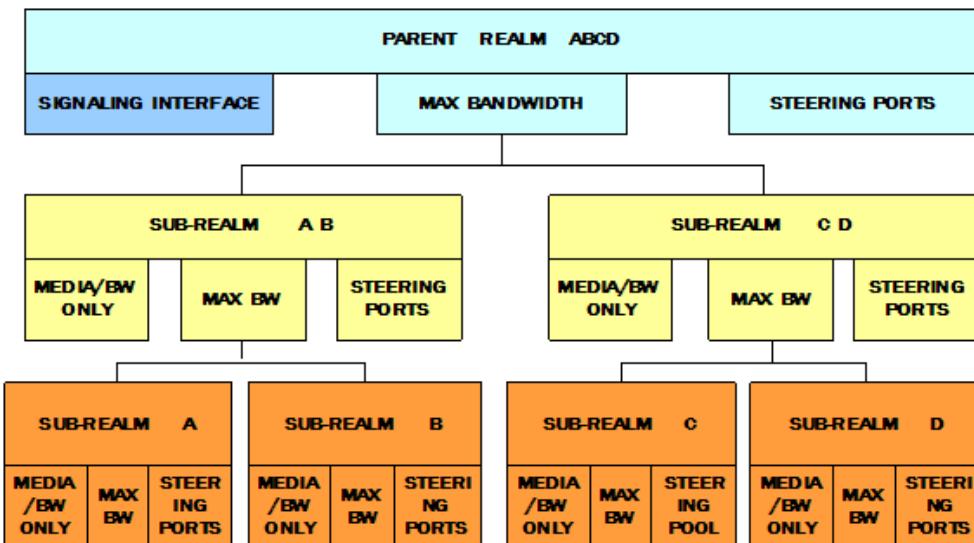
These benefits are achieved when you configure these types of realms:

- **Realm group**—A hierarchical nesting of realms identified by the name of the highest order realm.
- **Controlling realm**—A realms for which a signaling interface is configured. For example, you might configure these signaling interfaces in the following configurations: SIP-NAT, SIP port, H.323 stack, or MGCP. Typically, this is the highest order realm for the parent realm in a realm group.

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

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

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



## Configuring Nested Realms

When you are configuring nested realms, you can separate signaling and media by setting realm parameters in the SIP interface configuration, the H.323 stack configuration, and the steering ports configuration.

- The realm identifier you set in the SIP interface configuration labels the associated realm for signaling.
- The realm identifier you set in the H.323 stack configuration labels the associated realm for signaling.
- The realm identifier you set in the steering ports configuration labels the associated realm for media.

For MGCP, as explained below, you set a special option that enables nested realm use.

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

In the SIP port configuration for the SIP interface and in the H.323 stack configuration, you will find an allow anonymous parameter that allows you to set certain access control measures. The table below outlines what each parameter means.

**Table 1: Allow Anonymous Parameters**

| 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:

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

## Required Signaling Service Parameters

To configure nested realms, you need to set parameters in the realm configuration and in the configurations for the signaling protocols you want to use.

### To configure H.323 stack parameters for nested realms:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.  
ACMEPACKET(configure)# **session-router**

3. Type **h323** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#{}
```

4. Type **h323-stacks** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

From this point, you can configure H.323 stack parameters. To view all h323-stack configuration parameters, enter a ? at the system prompt.

5. **allow-anonymous**—Enter the admission control of anonymous connections accepted and processed by this H.323 stack. The default is **all**. The valid values are:

- **all**—Allow all anonymous connections
- **agents-only**—Only requests from session agents allowed
- **realm-prefix**—Session agents and address matching realm prefix

#### To configure MGCP for nested realms:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the **session-router** path.

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

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

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

```
ACMEPACKET(mgcp-config)#{}
```

4. You can either add support to a new MGCP configuration or to an existing MGCP configuration:

- 4a. If you do not currently have an MGCP configuration, you can add the option by typing options, a <Space> and then **nested-realm**.

```
ACMEPACKET(mgcp-config)#{ options nested-realm }
```

- 4b. Select the MGCP configuration so that you can add MGCP nested realm support to it. Then, to add this option to a list of options that you have already configured for the MGCP configuration, type **options** followed by a <Space>, the plus sign (+), and the **nested-realm** option.

```
ACMEPACKET(mgcp-config)#{ select
```

```
ACMEPACKET(mgcp-config)#{ options +nested-realm }
```

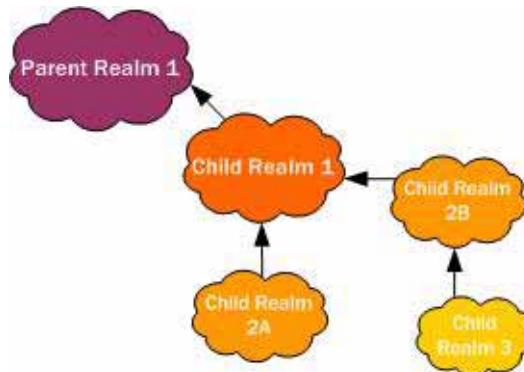
## Aggregate Session Constraints: Nested Realms

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.

### **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)# **sessl on-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 SBC supports TOS/DiffServ functions that allow you to

- Set up realm-based packet marking by media type, either audio-voice or video
- Set up realm-based packet marking for signaling, either SIP or H.323

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

### About TOS/DiffServ

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

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 and/or H.323
- One or more video media types for SIP and/or H.323
- Both audio and video media types for SIP and/or H.323

For all incoming SIP and H.23 requests, the media type is determined by negotiation or by preferred codec. SIP media types are determined by the SDP, and H.323 media types are determined by the media specification transmitted during call setup.

## Configuring Packet Marking by Media Type

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

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

```
name media policy name
tos-settings list of TOS settings
```

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

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

## ACLI Instructions and Examples: Packet Marking for Media

### 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)#
4. **name**—Enter the unique name of this media policy.
5. **tos-settings**—Enter the media type and TOS value for this media policy.

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

- 5a.**media-type**—Enter the media type you want to use.
- 5b.**tos-values**—Enter the TOS values to use for this media type. The valid range is:
- 0x00 to 0xFF

## 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# **config 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

### Configuring a Media Policy for Signaling Packet Marking

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

### 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# **config 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-settings**—Enter the media type and TOS value for this media policy.  
5a.**media-type**—Enter the media type you want to use.  
5b.**tos-values**—Enter the TOS values to use for this media type. The valid range is:

- 0x00 to 0xFF

## 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, this is the value to use.
  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# **config terminal**
2. Type **media-manager** and press <Enter>.   
ACMEPACKET(config)# **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(medi a-pol i cy)# **tos-setti ngs**
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-setti ngs)# **medi a-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-setti ngs)# **medi a-sub-type sipp**
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 ("").  
ACMEPACKET(tos-setti ngs)# **medi a-attribut es "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-setti ngs)# **tos-val ue 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 SBC. 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 SBC 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 SBC 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 SBC. For example, if your Net-Net SBC 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 SBC will never use.

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

| Parameter  | Description                                                                                                                                                                |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IP address | IPv4 address of the steering pool.                                                                                                                                         |
| start port | Port number that begins the range of ports available to the steering pool.<br>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.<br>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

### To configure a steering pool:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
**ACMEPACKET# config terminal**
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.   
**ACMEPACKET(config)# media-manager**
3. Type **steering-pool** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
**ACMEPACKET(media-manager)# steering-pool**  
**ACMEPACKET(steering-pool) #**
4. **ip-address**—Enter the target IPv4 address of the steering pool in IP address format. For example:  
**192.168.0.11**
5. **start-port**—Enter the start port value that begins the range of ports available to this steering pool. The default is 0. The valid range is:
  - Minimum—0
  - Maximum—65535
 You must enter a valid port number or the steering pool will not function properly.
6. **end-port**—Enter the end port value that ends the range of ports available to this steering pool. The default is 0. The valid range is:
  - Minimum—0

- Maximum—65535

You must enter a valid port number or the steering pool will not function properly.

7. **realm-id**—Enter the realm ID to identify the steering pool's realm, following the name format. The value you enter here must correspond to the value you entered as the identifier (name of the realm) when you configured the realm. For example:

**peer-1**

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

The following example shows the configuration of a steering pool that

**steering-pool**

|                           |                     |
|---------------------------|---------------------|
| <b>ip-address</b>         | 192.168.0.11        |
| <b>start-port</b>         | 20000               |
| <b>end-port</b>           | 21000               |
| <b>realm-id</b>           | peer-1              |
| <b>Last-modified-date</b> | 2005-03-04 00:35:22 |

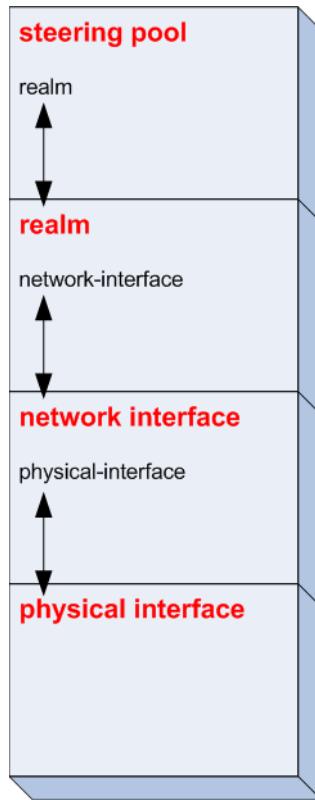
## Multiple Interface Realms

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 SBC 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 SBC to create a standard one-realm to one-network interface configuration.

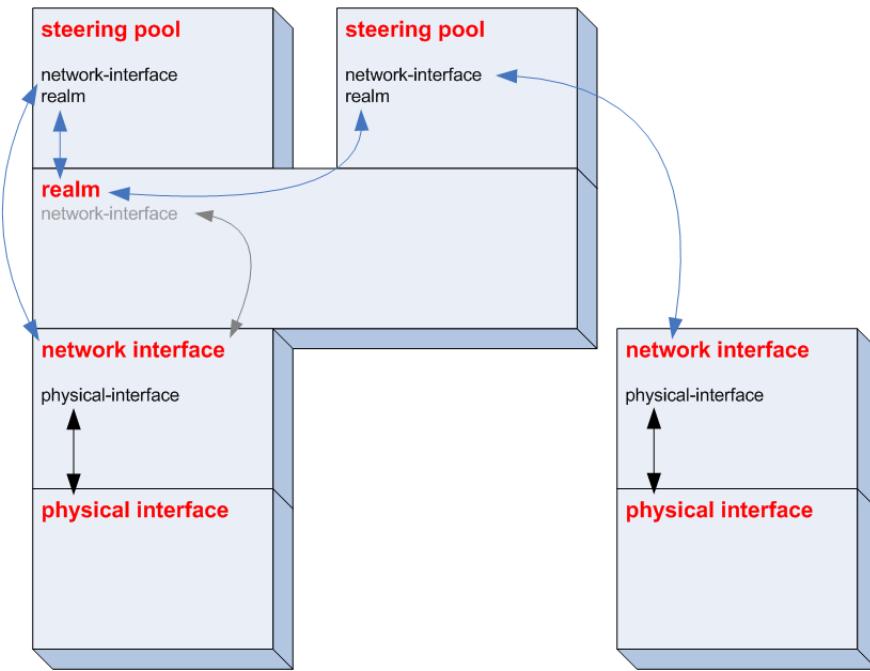
### How It Works

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

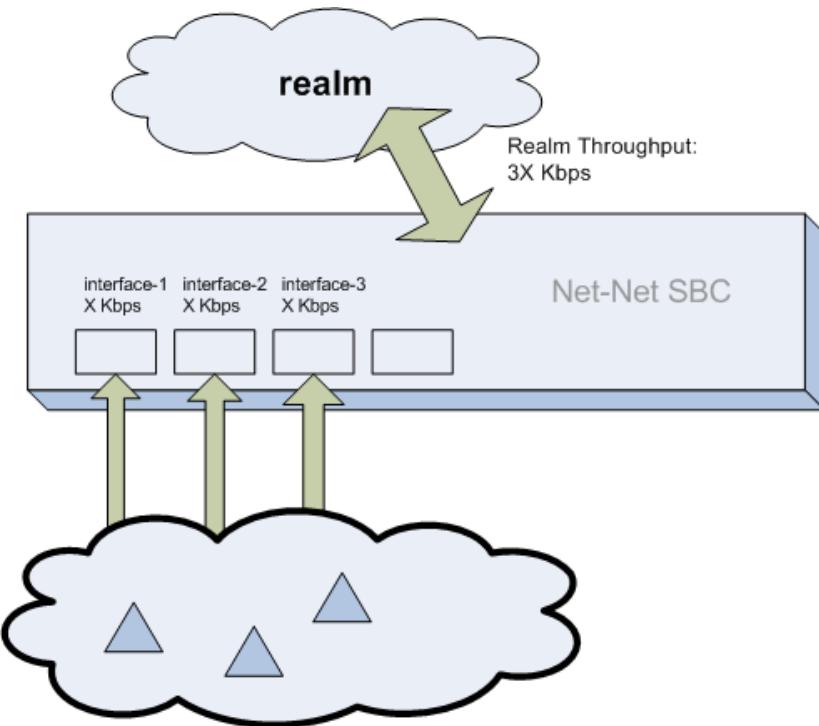


In this model, one (non-channelized) network interface exists on a physical interface. One realm exists on one network interface. One or more steering pools can exist on one realm. Within each higher level configuration element exists a parameter that references a lower level configuration element in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

### Creating a List of Network Interfaces for the Realm

This section explains how to configure your Net-Net SBC to use multiple interface realms.

You must first configure multiple physical interfaces and multiple network interfaces on your Net-Net SBC.

#### To configure the realm configuration for multi-interface realms.

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **confi gure terminal**
2. Type **media-manager** and press <Enter> to access the media-manager path.   
ACMEPACKET(configure)# **medi a-manager**
3. Type **realm-config** and press <Enter>. The system prompt changes.   
ACMEPACKET(medi a-manager)# **real m-confi g**  
ACMEPACKET(real m-confi g)#
 

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# **confi gure terminal**
2. Type **media-manager** and press <Enter> to access the media-manager path.   
ACMEPACKET(configure)# **medi a-manager**

3. Type **steering-pool** and press <Enter>. The system prompt changes.  
ACMEPACKET(medi a-manager)# **steeri ng-pool**  
ACMEPACKET(steeri ng-pool )#  
From this point, you can configure steering pools which collectively bridge the multiple network interfaces they are connected to.
4. **ip-address**—Enter the IP address of the first steering pool on the first network interface.  
This IP address must correspond to an IP address within the subnet of a network interface you have already configured.  
This IP can not exist on a network interface other than the one you configure in the **network-interface** parameter.
5. **start-port**—Enter the beginning port number of the port range for this steering pool. The default is 0. The valid range is:
  - Minimum—0
  - Maximum—65535
6. **end-port**—Enter the ending port number of the port range for this steering pool. The default is 0. The valid range is:
  - Minimum—0
  - Maximum—65535
7. **realm-id**—Enter the name of the realm which this steering pool directs its media traffic toward.
8. **network-interface**—Enter the name of the network interface you want this steering pool to direct its media toward. This parameter will match a **name** parameter in the network-interface configuration element. If you do not configure this parameter, you can only assign a realm to a single network interface, as the behavior was in all SD Software releases pre- 2.1.
9. Create additional steering pools on this and on other network interfaces as needed. Remember to type **done** when you are finished configuring each new steering pool.

## Media over TCP

The Net-Net SBC now supports RFC 4145 (TCP-Based Media Transport in the SDP), also called TCP Bearer support. Media over TCP can be used to support applications that use TCP for bearer path transport.

RFC 4145 adds two new attributes, *setup* and *connection*, to SDP messages. The *setup* attribute indicates which end of the TCP connection should initiate the connection. The *connection* attribute indicates whether an existing TCP connection should be used or if a new TCP connection should be setup during re-negotiation. RFC 4145 follows the offer/answer model specified in RFC3264. An example of the SDP offer message from the end point 192.0.2.2 as per RFC4145 is as given below:

```
m=video 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 SBC does not take an active part in the application-layer communication between each endpoint.

The Net-Net SBC 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 SBC's support of RFC 4145.

1. The Net-Net SBC 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 SBC does not inspect the *setup* and *connection* attributes in the SDP message since the TCP packets transparently pass through the Net-Net SBC. These SDP attributes are forwarded to the other endpoint. It is the other endpoint's responsibility to act accordingly.
4. After the Net-Net SBC 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

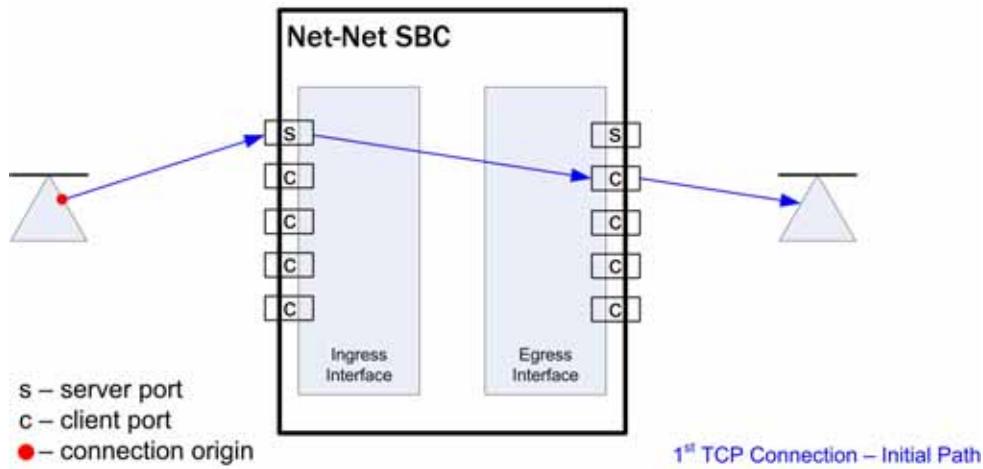
When a call is first set up, the Net-Net SBC inspects the SDP message's m-line to see if any media will be transported via TCP. If the SDP message indicates that some content will use TCP, the Net-Net SBC allocates a configured number of steering ports for the media-over-TCP traffic. These TCP media 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 SBC 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 SBC 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 SBC counts this number as inclusive of the server port. Therefore if you want the Net-Net SBC 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 SBC through a client port on the egress interface toward an endpoint:

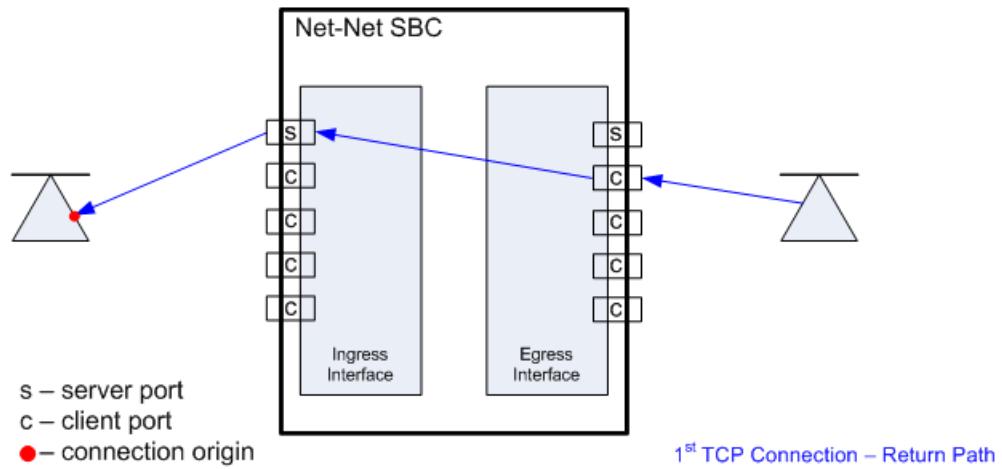
## TCP Connection 1 - Eastward Path



The endpoint responds back to the client port on the egress interface. This message traverses the Net-Net SBC and is forwarded out of the server port on the ingress

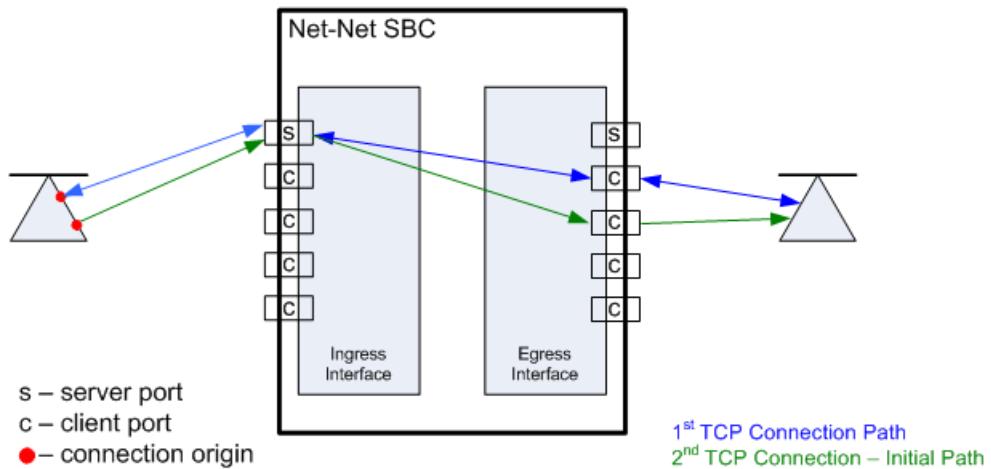
interface where the initial packet was sent. The remainder of the TCP flow uses the server and client port pair as a tunnel through the Net-Net SBC:

### TCP Connection 1 - Westward Path



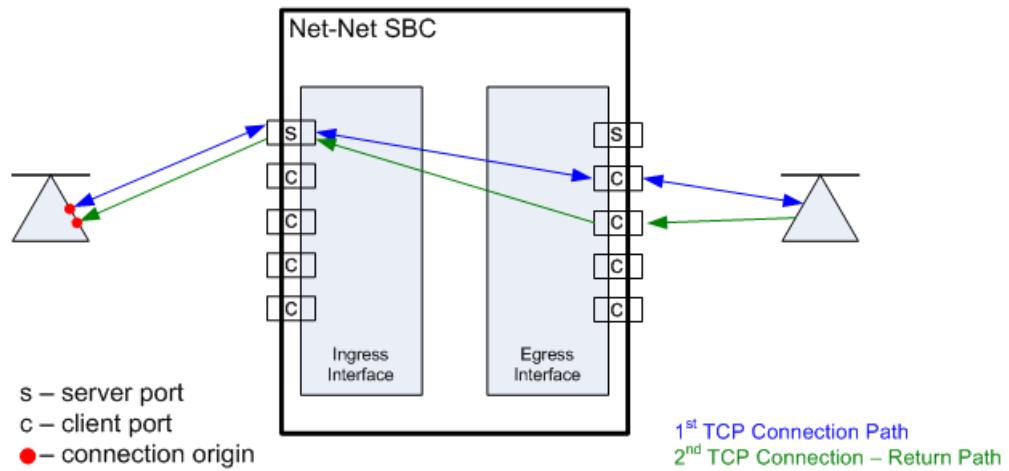
When the second TCP connection is set up in the same direction as in the first example, the first packet is still received on the server port of the ingress interface. The next unused client port is chosen for the packet to exit the Net-Net SBC:

### TCP Connection 2 - Eastward Path



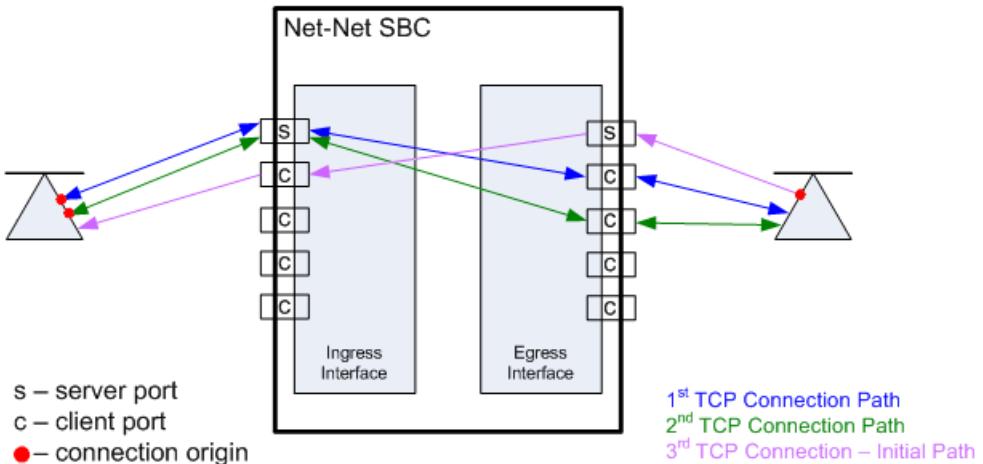
The response takes the same path back to the caller. The remainder of the second TCP connection uses this established path:

### TCP Connection 2 - Westward Path



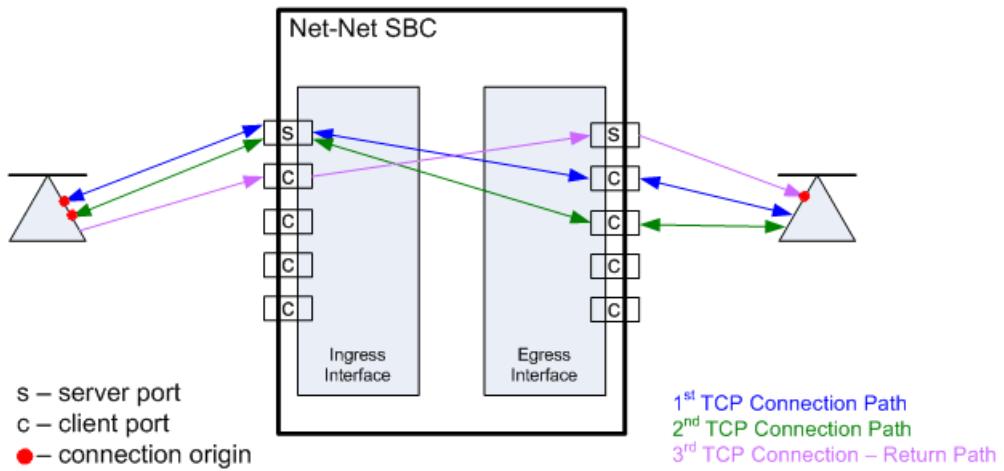
When the callee initiates a TCP connection, it must send its initial traffic to the server port on its Net-Net SBC ingress interface. The packet is forwarded out of the first free client port on the egress side of this TCP connection toward the caller.

### 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



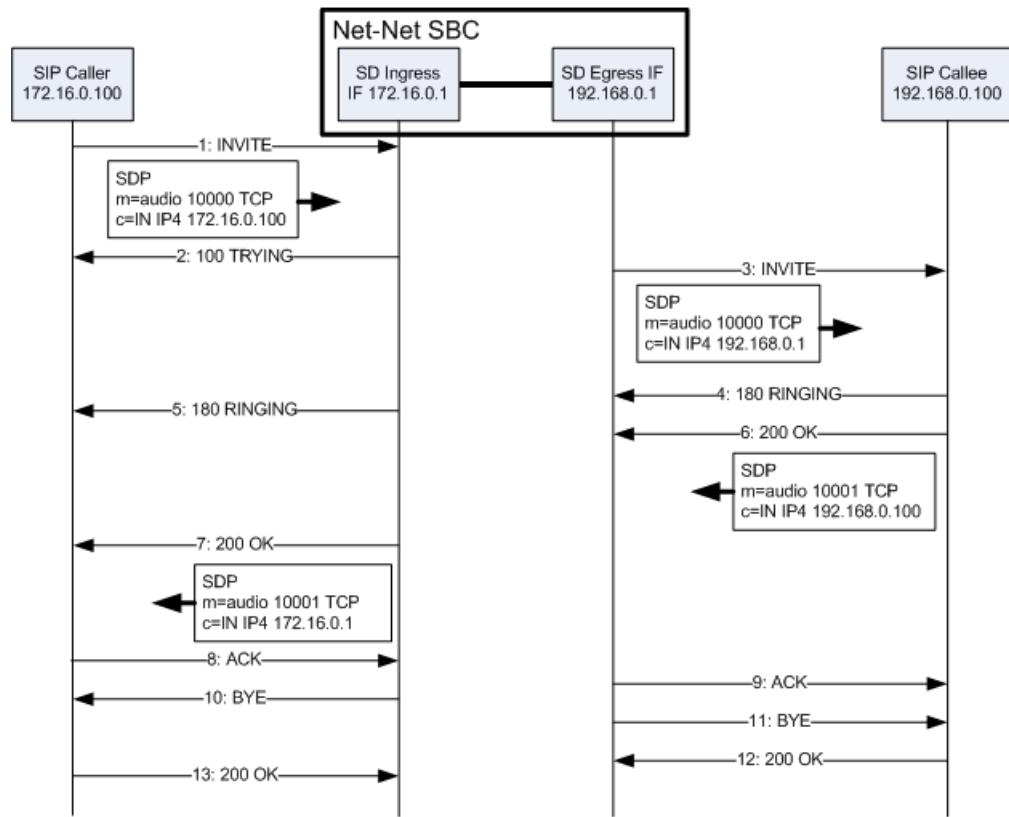
The Net-Net SBC can support a total of eight media-over-TCP connections per call. A maximum of 4 connections are supported as initiated from each side of the call.

### 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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# **config**ure **termi**nal
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.  
ACMEPACKET(config)# **medi**a-**manager**  
ACMEPACKET(media-manager)#[/ol>

3. Type **media-manager** and press <Enter> to begin configuring media over TCP.  
 ACMEPACKET(medi a-manager)# **medi a-manager**  
 ACMEPACKET(medi a-manager-confi g)#
4. **tcp-number-of-ports-per-flow**—Enter the number of ports, inclusive of the server port, to use for media over TCP. The total number of supported flows is this value minus one. The default is 2. The valid range is:
  - Minimum—2
  - Maximum—5
 ACMEPACKET(real m-confi g)# **tcp-number-of-ports-per-flow 5**
5. **tcp-initial-guard-timer**—Enter the maximum time in seconds allowed to elapse between the initial SYN packet and the next packet in a media-over-TCP flow. The default is 300. The valid range is:
  - Minimum—0
  - Maximum—999999999
 ACMEPACKET(real m-confi g)# **tcp-initial-guard-timer 300**
6. **tcp-subsq-guard-timer**—Enter the maximum time in seconds allowed to elapse between all subsequent sequential media-over-TPC packets. The default is 300.
  - Minimum—0
  - Maximum—999999999
 ACMEPACKET(real m-confi g)# **tcp-subsq-guard-timer 300**
7. **tcp-flow-time-limit**—Enter the maximum time in seconds that a media-over-TCP flow can last. The default is 86400. The valid range is:
  - Minimum—0
  - Maximum—999999999
 ACMEPACKET(real m-confi g)# **tcp-flow-time-limit 86400**

## Restricted Media Latching

The restricted media latching feature 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. It sets the source address and address prefix bits in the flow. It also parses and loads the source flow address into the MIBOCO messages. After receiving the calling SDP, the 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. In addition, sipd sends the information for both the incoming and outgoing flows. After receiving SDP from the called side sipd, the Net-Net SBC sends information for both flows to the MBCD so that the correct NAT entries are installed.

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(real-m-config)#
    4. Select the realm where you want to apply this feature.  
ACMEPACKET(real-m-config)# **select**  
identifier:  

|                      |         |
|----------------------|---------|
| 1: Acme_Realm <none> | 0.0.0.0 |
| 2: MGCP_Realm <none> | 0.0.0.0 |
| 3: H323REALM <none>  | 0.0.0.0 |

  
selection: 1

```
ACMEPACKET(real m-confi g)#

```

5. **restricted-latching**— Enter the restricted latching mode. The default is **none**. The valid values are:
  - **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. This field will be used only if the restricted-latching is used. The default is **32**. When this value is used, the complete IP address is matched. The valid range is:
  - Minimum—1
  - Maximum—32

## Media Release Across SIP Network Interfaces

---

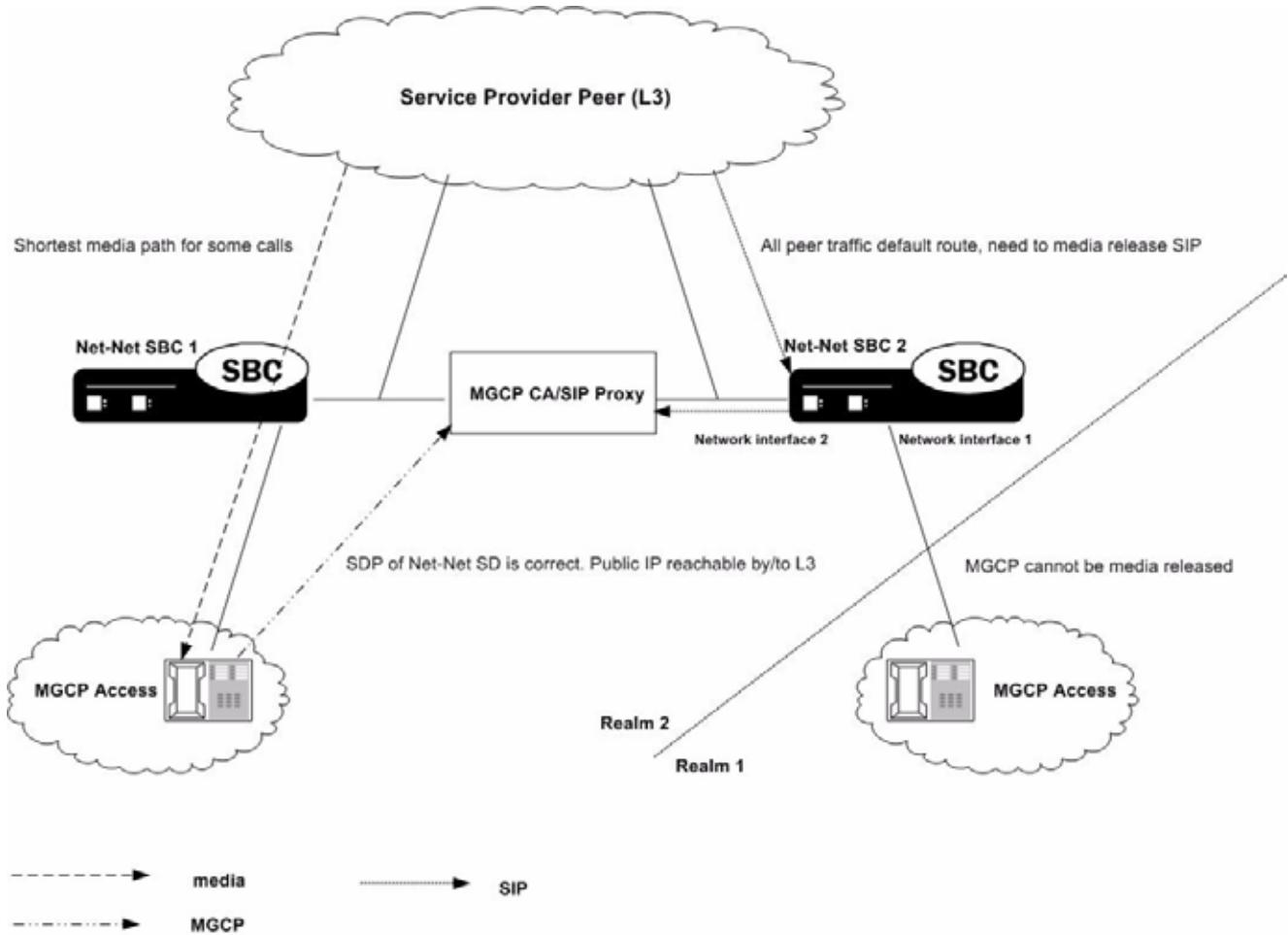
This feature lets the Net-Net SBC release media between two SIP peers, between two realms on two network interfaces of the same Net-Net SBC. Use this feature when you want the Net-Net SBC to release media for specific call flows, regardless of the attached media topology.

### Example

You can have two or more Net-Net SBCs with MGCP realms, performing MGCP signaling, media, and NATing to the MGCP call agent. The call agent signals SIP to peers (Level 3) for off-net calls, always through a default Net-Net SBC route. In many cases, the Net-Net SBC being used for SIP call routing (SBC2) is not the same Net-Net SBC where the MGCP endpoint resides (SBC1). In addition, a more direct media path exists between the MGCP-served Net-Net SBC (SBC1) and Level-3. The SDP provided by the Net-Net SBC MGCP ALG (SBC1) is public and can be routed to Level 3. However, the SIP default route Net-Net SBC (SBC2) is also an MGCP ALG and cannot have global media release. It must keep media management for MGCP.

SIP can also arrive from other Net-Net SBCs (or perhaps go out through them in the future). The Net-Net SBC must be able to perform similar media release for SIP while managing media for MGCP or access SIP realms.

In the following diagram, the access realms for endpoints are currently MGCP, with the expectation they will be migrated to SIP in the future.



## ACLI Instructions and Examples

To configure media release across network interfaces:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **config terminal**
  2. Type **media-manager** and press <Enter> to access the media-level configuration elements.   
ACMEPACKET(config)# **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(real-m-config)#
      4. Select the realm where you want to apply this feature.   
ACMEPACKET(real-m-config)# **select**  
**identifier:**  
1: Acme\_Real m <none> 0.0.0.0  
2: MGCP\_Real m <none> 0.0.0.0

```
3: H323REALM <none> 0.0.0.0
```

```
selection: 1
ACMEPACKET(realm-config) #
```

5. **mm-in-system**—Enable to release media between two SIP peers, between two realms on two network interfaces of the same Net-Net SBC. Disable to always release the media, regardless of interface and realm. The default is **enabled**. The valid values are:
  - enabled | disabled

## Media Release Behind the Same IP Address

---

The media management behind the same IP feature lets the Net-Net SBC release media when two endpoints are behind the same IP address, in the same realm. Using this feature prevents the media for intra-site calls from going through the Net-Net SBC. You can use this feature for both hosted NAT traversal (HNT) and non-HNT clients. It works with NATed endpoints and for non-NATed ones that are behind the same IP.

### Additional Media Management Options

Additional media management options include:

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

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

```
ACMEPACKET(medi a-manager)# real m-confi g
ACMEPACKET(real m-confi g)#

```

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

4. **mm-in-realm**—Enable if you plan to use **mm-same-ip**. If this parameter is disabled, the **mm-same-ip** parameter is ignored. If you set this to **enabled** and **mm-same-ip** to **disabled**, media is managed in the realm but released if the two endpoints are behind the same IP address. The default is **disabled**. The valid values are:
  - enabled | disabled
5. **mm-same-ip**—Enable if you want media to go through this Net-Net SBC, if **mm-in-realm** is **enabled**. When **disabled**, the media will not go through the Net-Net SBC for endpoint that are behind the same IP. The default is **enabled**. The valid values are:
  - enabled | disabled

## Bandwidth CAC for Media Release

---

The bandwidth CAC for media release feature adds per-realm configuration that determines whether or not to include inter-realm calls in bandwidth calculations. When you use this feature, the 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(medi a-manager)#`
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.  
`ACMEPACKET(medi a-manager)# real m-confi g`  
`ACMEPACKET(real m-confi g)#`
4. Select the realm where you want to want to add this feature.  
`ACMEPACKET(real m-confi g)# 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 of them:

- Is directly addressable at the same IP address as a NAT device, but is not behind a NAT device
- Is at the same IP address of a NAT device the other endpoint is behind

You enable this feature on a per-realm basis by setting an option in the realm configuration.

When this option is not set, 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# config terminal
ACMEPACKET(config)#
```

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

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

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

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

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

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **release-media-at-same-nat** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

## Media Release Behind the Same NAT IP Address

---

You can now configure your 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 enabled, then **mm-in-realm** or **mm-in-network** can both be `enabled`.

### To set the option that enables media release behind the same IP address:

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

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

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

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

```

3. Type `realm-config` and press <Enter>.

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

```

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

4. **options**—Set the options parameter by typing `options`, a <Space>, the option name **release-media-at-same-nat** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(realm-config)# options +release-media-at-same-nat
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

## Codec Reordering

Certain carriers deploy voice services where their peering partners do not use the carriers’ preferred codecs. The Net-Net SBC can now reorder the codecs so that the preferred one is selected first.

Take the example of a carrier that deploys a voice service using G.729 rather than G.711. If that carrier has a peering partner providing call origination for the VoIP customers with G.711 used as the preferred codec, there can be issues with codec selection.

The Net-Net SBC resolves this issue by offering its codec reordering feature. Enabled for realms and session agents, this feature gives the Net-Net SBC the ability to reorder the default codec in an SDP offer to the preferred codec before it forwards the offer to the target endpoint. When you enable this feature, you increase the probability that the target endpoint will choose the preferred codec for its SDP answer, thereby avoiding use of the undesired codec.

## How It Works

You enable codec reordering feature by setting the `preferred-codec=X` (where X is the preferred codec) option in the realm and session agent configurations. You set it in the realm from which the Net-Net SBC receives SDP offers (in requests or responses), and for which the media format list needs to be reordered by the Net-Net SBC prior to being forwarded. To configure additional codec ordering support for cases when a response or request with an SDP offer is from a session agent, you can set this option in the session agent configuration.

If you enable the option, the Net-Net SBC examines each SDP media description before it forwards an SDP offer. And if necessary, it performs reordering of the media format list to designate that the preferred codec as the default.

The Net-Net SBC determines preferred codecs in the following ways:

- If the response or request with an SDP offer is from a session agent, the Net-Net SBC determines the preferred codec by referring to the session agent configuration. You set the preferred codec for a session agent by configuring it with the `preferred-codec=X` option.
- If the response or request with an SDP offer is not from a session agent or is from a session agent that does not have the `preferred-codec=X` option configured, the Net-Net SBC determines the preferred codec by referring to the `preferred-codec=X` option in the realm.
- If the Net-Net SBC cannot determine a preferred codec, it does not perform codec reordering.

The way that the Net-Net SBC performs codec reordering is to search for the preferred codec in the SDP offer's media description (`m=`) line, and designate it as the default codec (if it is not the default already). After it marks the preferred codec as the default, the Net-Net SBC does not perform any operation on the remaining codecs in the media format list. Note that the Net-Net SBC performs codec reordering on the media format list only. If the `rtpmap` attribute of the preferred codec is present, the Net-Net SBC does not reorder it.

## Preferred Codec Precedence

When you configure preferred codecs in session agents or realms, be aware that the codec you set for a session agent takes precedence over one you set for a realm. This means that if you set preferred codecs in both configurations, the one you set for the session agent will be used.

In the case where the Net-Net SBC does not find the session agent's preferred codec in the SDP offer's media format list, then it does not perform codec reordering even if the media format list contains the realm's preferred codec.

## ACLI Instructions and Examples

When you configure codec ordering, the codec you set in either the session agent or realm configuration must match the name of a media profile configuration. If your configuration does not use media profiles, then the name of the preferred codec that you set must be one of the following:

- PCMU
- G726-32
- G723
- PCMA
- G722
- G728

- G729

**Note:** If you configure this feature for a session agent, you must configure it for the associated realm as well. Otherwise, the feature will not work correctly.

## Setting a Preferred Codec for a Realm

### To set a preferred codec for a realm configuration:

These instructions assume that you want to add this feature to a realm that has already been configured.

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter>.   
ACMEPACKET(configure)# **media-manager**  
ACMEPACKET(media-manager)#
  3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(media-manager)# **realm-config**  
ACMEPACKET(real-m-config)#
    4. Select the realm where you want to apply this feature.   
ACMEPACKET(real-m-config)# **select**  
identifier:  
1: public medi a2: 0 0.0.0.0  
2: private medi a1: 0 0.0.0.0  
  
selection: 1  
ACMEPACKET(real-m-config)#
    5. **options**—Set the **options** parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**preferred-codec=X**), and then press <Enter>. X is the codec that you want to set as the preferred codec.   
ACMEPACKET(real-m-config)# **options +preferred-codec=PCMU**  
If you type **options preferred-codec=X**, you will overwrite any previously configured options. In order to append the new option to the **realm-config**'s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
    6. Save and activate your configuration.

## Setting a Preferred Codec for a Session Agent

### To set a preferred codec for a session agent configuration:

These instructions assume that you want to add this feature to a session agent that has already been configured.

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter>.   
ACMEPACKET(configure)# **session-router**  
ACMEPACKET(session-router)#
  3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

- ```

ACMEPACKET(session-router)#
ACMEPACKET(session-agent)#
4. Select the session agent where you want to apply this feature.
ACMEPACKET(session-agent)# select
<hostname>
1: acmepacket.com realm=      ip=
2: sessionAgent2 realm=tester ip=172.30.1.150

selection:
selection: 1
ACMEPACKET(session-agent)#
5. options—Set the options parameter by typing options, a <Space>, the option name preceded by a plus sign (+) (preferred-codec=X), and then press <Enter>. X is the codec that you want to set as the preferred codec.
ACMEPACKET(session-agent)# options +preferred-codec=PCMU
If you type options preferred-codec=X, you will overwrite any previously configured options. In order to append the new option to the session agent's options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

```

Media Profiles Per Realm

For different codecs and media types, you can set up customized media profiles that serve the following purposes:

- Police media values
- Define media bandwidth policies
- Support H.323 slow-start to fast-start interworking

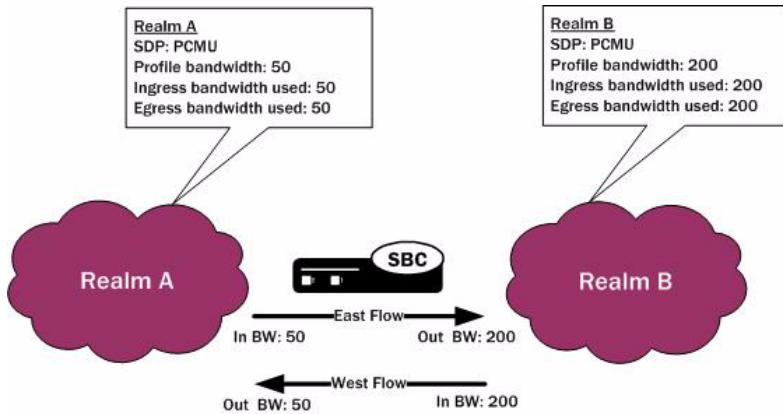
You can use media policies globally for the Net-Net SBC, or—starting with Release C6.1.0—you can configure them for application on a per-realm basis. For a realm, you can configure a list of media profiles you want applied. The Net-Net SBC matches the value you set for the **match-media-profiles** parameter, 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 call exceeds bandwidth on either the ingress or egress side, the Net-Net SBC rejects the call. You can also use per-user CAC, which limits the maximum bandwidth from the east and west flows for both the TO and FROM users.

When you apply media profiles to a realm, the 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 on 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>. Releases C6.1.0 and later accept the subname so you can configure multiple media profile for the same codec; the codec **name** customarily serves as the name value for a media profile configuration.

About Wildcarding

You can wildcard both portions (name and subname) of this value:

- When you wildcard the **name** portion of the value, you can provide a specific subname that the 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.

You can also enter the name value on its own, or wildcard the entire value. Leaving the subname value empty is also significant in that it allows the realm to use all media profile that have no specified **subname**. However, you cannot leave the **name** portion of the value unspecified (as all media profiles are required to have names).

Consider the examples in the following table:

Syntax	Example Value	Description
<name>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <name>::*.
<name>::	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<name>::<subname>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
*	*	Matches anything, but does not have to be a defined media profile.

Syntax	Example Value	Description
::	*::*	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

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# config terminal
ACMEPACKET(config)#End
```

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

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

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)#End
```

4. **match-media-profiles**—In the form **<name>::<subname>**, enter the media profiles you would like applied to this realm. These values correspond to the name and subname parameters in the media profile configuration. You can wildcard either of these portions of the value, or you can leave the **<subname>** portion empty. Refer to the [About Wildcarding \(211\)](#) section above for more information about syntax, wildcarding, and empty values.

This parameter has no default.

5. Save and activate your configuration.

Multiple Media Profiles

You can use the media profiles configuration to set up:

- One media profile for a particular SIP SDP encoding (such as G729), where the name of the profile identifies it uniquely. This behavior is your only option in Net-Net OS release prior to Release C6.1.0.
- Multiple media profiles for the same SIP SDP encoding. Available in Release C6.1.0 and forward, you can create multiple media profiles for the same encoding. To do so, you add a subname to the configuration, thereby identifying it uniquely using two pieces of information rather than one.

The sections below provide two descriptions of deployments where using multiple profiles for the same codec would solve codec and packetization problems for service providers.

Use Case 1

Service Provider 1 peers with various carriers, each of which uses different packetization rates for the same codec. For example, their Peer 1 uses 10 milliseconds G.711 whereas their Peer 2 uses 30 milliseconds for the same codec. The difference in rates produces a difference in bandwidth consumption—resulting in a difference in SLA agreements and in Net-Net SBC call admission control (CAC) and bandwidth policing. Service Provider 1 uses the Net-Net SBC's media profile configuration parameters to determine CAC (**req-bandwidth**) and bandwidth policing (**avg-rate-limit**). Because this service provider's peers either do not use the SDP p-time attribute or use it inconsistently, it is difficult to account for bandwidth use. And so it is likewise difficult to set up meaningful media profiles.

The best solution for this service provider—given its traffic engineering and desire for the cleanest routing and provisioning structures possible—is to define multiple media profiles for the same codec.

Use Case 2

Service Provider 2 supports H.263 video, for which the Net-Net SBC offers a pre-provisioned media profile with a set bandwidth value. And yet, H.263 is not a codec that has a single bandwidth value. Instead, H.263 can have different bandwidth values that correspond to various screen resolution and quality. While it is true that the Net-Net SBC can learn the requisite bandwidth value from SDP, not all SDP carries the bandwidth value nor do system operators always trust the values communicated.

Configuring multiple media profiles for the same codec (here, H.263) helps considerably with this problem—and moves closer to complete solution. Service Provider 2 can configure H.263 media profiles capable of handling the different bandwidth values that might appear.

ACLI Instructions and Examples

Configuring the **subname** parameter in the media profiles configuration allows you to create multiple media profiles with the same name.

To configure the subname parameter for a media profile:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **media-profile** and press <Enter>. If you are adding this feature to a pre-existing media profile configuration, you will need to select and edit your media profile.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **subname**—Enter the subname value for this media profile. Information such as the rate or bandwidth value make convenient subname values. For example, you might set the **name** of the media profile as PCMU and the **subname** as 64k.

This parameter is not required and has no default.

5. Save and activate your configuration.

Peer-to-Peer MSRP TCP Stitching

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.

Introduction

This chapter explains how to configure the Net-Net SBC 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 SBC and SIP

This section describes the Net-Net SBC's support of SIP. It provides the basic information you need to understand before you configure the Net-Net SBC 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 SBC operates as a back-to-back user agent (B2BUA) within the following two basic service models:

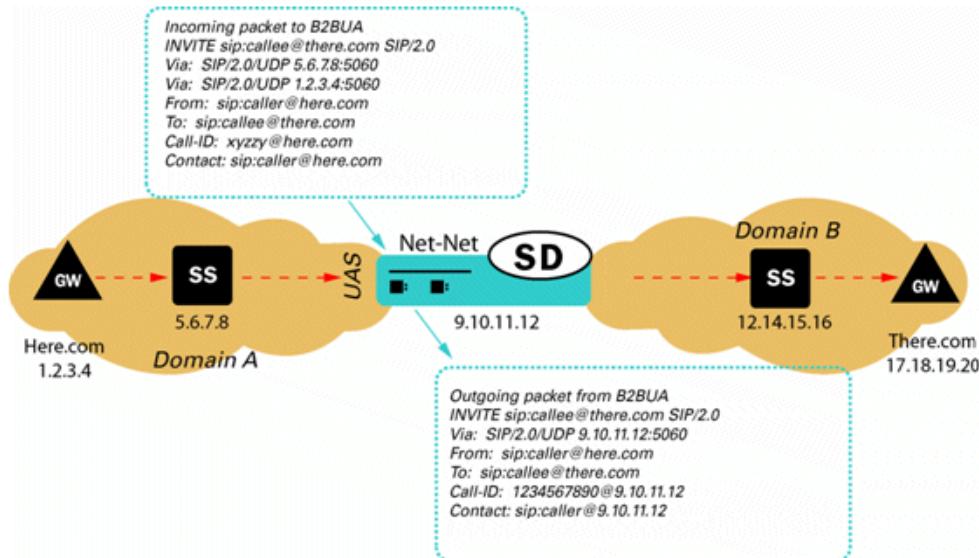
- peering
- hosted IP services

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

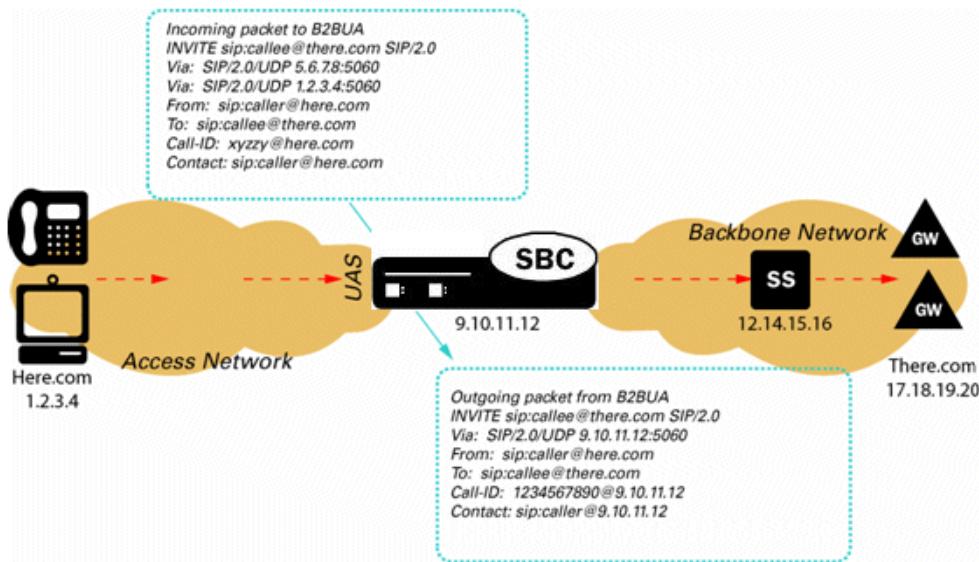
The Net-Net SBC operates as a SIP B2BUA. It terminates SIP sessions and re-originate them as new sessions as they are routed through the Net-Net SBC. For each session, it establishes NAPT translations and re-writes SDP to allow all session related media to be routed through the Net-Net SBC. It generates 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 SBC 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 SBC 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 SBC 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 SBC. Also, all calling features such as caller ID, call waiting, three-way calling, and call transfer are all supported transparently through the Net-Net SBC.



SIP B2BUA and L3/L5 NAT

For each SIP session, the Net-Net SBC establishes NAPT translations and re-writes SDP to route all session related media through the Net-Net SBC. These actions make the Net-Net SBC look like a SIP gateway. Also, the Net-Net SBC support of multiple SIP interfaces associated with different network interfaces makes it appear as multiple virtual SIP gateways.

This functionality enables the Net-Net SBC 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 SBC receives and sends SIP messages. You can define a SIP interface for each network or realm to which the Net-Net SBC is connected. SIP interfaces support both UDP and TCP transport, as well as multiple SIP ports (transport addresses). The SIP interface's SIP NAT function lets Hosted NAT Traversal (HNT) be used in any realm.

SIP INVITE Message Processing

When the session agent element on the softswitch side of the message flow (ingress session agent) has the gateway contact parameter configured as an option, the Net-Net SBC looks for the URI parameter (as defined by the gateway contact parameter) in the Request-URI and decodes the gateway address.

Example

The following example shows a SIP INVITE message from a softswitch to a Net-Net SBC.

```
INVITE si p: 05030205555@ss-si de-ext-address; gateway=encoded-gw-address
From: "Anonymous" <si p: anonymous@anonymous. i nval i d>; tag=xxxx
To: <si p: 05030205555@ss-si de-ext-address; user=phone>
```

The following example shows a SIP INVITE message from a Net-Net SBC to a gateway.

```
INVITE si p: 05030205555@gw-i p-address SIP/2.0
From: "Anonymous" <si p: anonymous@anonymous. i nval i d>; tag=SDxxxx-xxxx
To: <si p: 05030205555@ hostpart; user=phone>
```

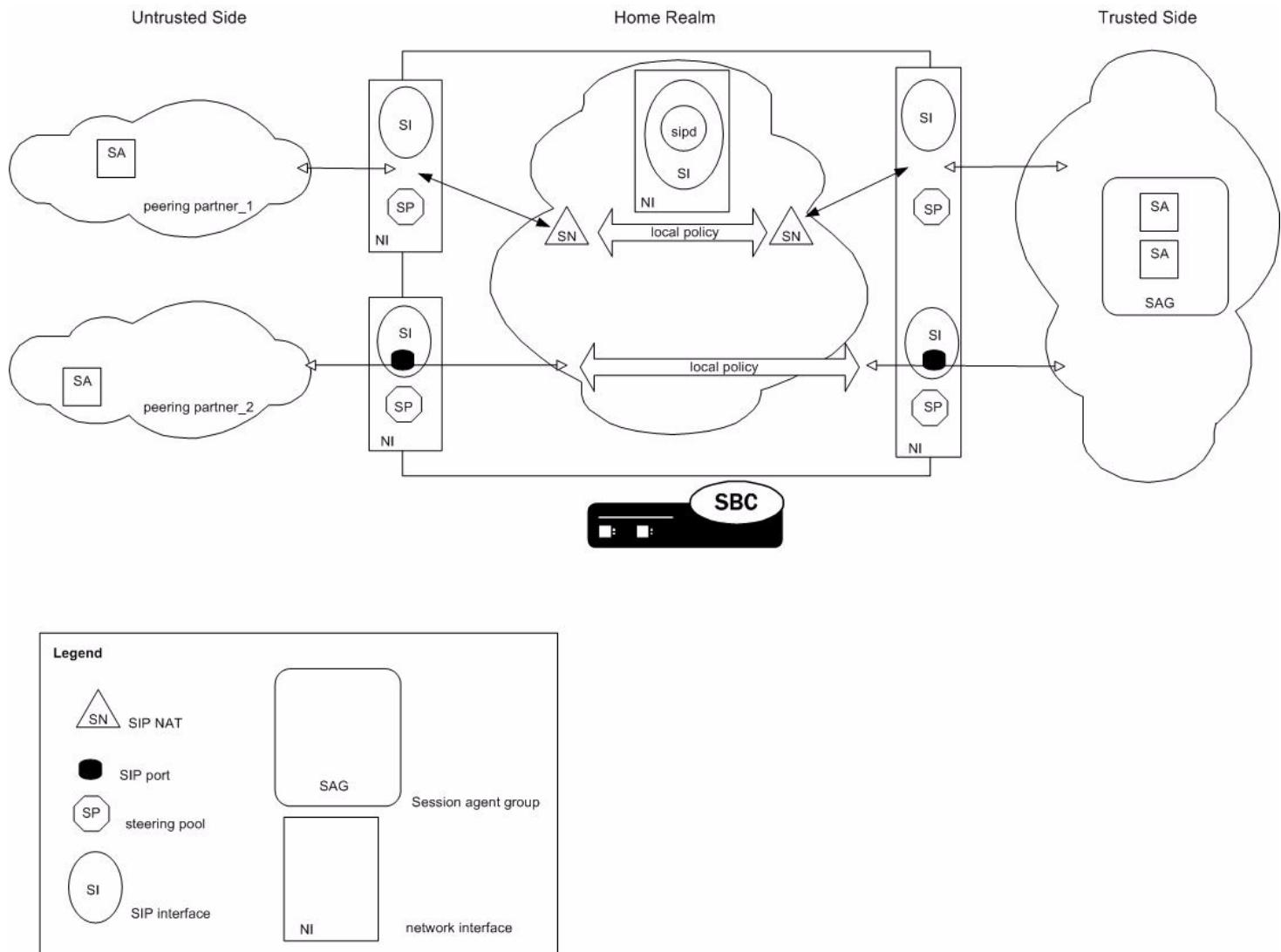
The Net-Net SBC converts the `hostpart` in the `To` header except in the following scenarios:

- when the original `hostpart` value received is a Fully Qualified Domain Name (FQDN)
- when the Net-Net SBC is configured *not* to NAT the `To` headers.

Acme Packet recommends configuring the Net-Net SBC to NAT the `To` headers to ensure the security of protected addresses. Otherwise, the outgoing `hostpart` is set to the SIP NAT's external proxy address for the SIP NAT's external realm.

Configuring the Net-Net SBC for SIP Signaling

This section contains a diagram of a B2BUA peering environment that illustrates the Net-Net SBC components you need to configure.



Refer to the following sections for details about configuring the Net-Net SBC 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 SBC and is where the Net-Net SBC'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 SBC. 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 \(257\)](#) 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 SBC connects.

Home Realm's Purpose

A home realm is required because the home address for SIP NATs is used to create a unique encoding of SIP NAT cookies. You can define the home realm as a network internal to the Net-Net SBC, which eliminates the need for an actual home network connected to the Net-Net SBC. 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 SBC 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# config terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

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

3. Type **sip-config** and press <Enter>. The system prompt changes.

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

```
ACMEPACKET(sip-config)#

```

From this point, you can configure SIP configuration parameters. To view all **sip-config** parameters, enter a **?** at the system prompt.

4. **home-realm-id**—Enter the name of the realm you want to use for the realm ID. For example, **acme**.

The name of the realm must correspond to the identifier value you entered when you configured the realm.

5. **egress-realm-id**—*Optional*. Enter the egress realm ID to define the default route for SIP requests addressed to destinations outside the home realm's address prefix.

If you enter a value for this optional field, it must correspond to the identifier value you entered when you configured the realm.

Note: You should leave this parameter blank for access/backbone applications. When left blank, the realm specified in the **home-realm-id** parameter is used by default.

6. **nat-mode**—Indicate the SIP NAT mode. The default is **none**. The valid values are:

- **public**—Indicates the subnet defined in the **addr-prefix-id** field of the home realm is public and the subnet defined in the **addr-prefix-id** field of all external realms identified in the SIP NAT are private networks. IPv4 addresses are encoded in SIP messages received from the external realm defined by the SIP NAT. The IPv4 addresses are decoded in messages that are sent to the realm.
- **private**—Indicates the subnet defined in the **addr-prefix-id** field of the home realm is private and the subnet defined in the **addr-prefix-id** field of all external realms identified in the SIP NAT are public networks. IPv4 addresses are encoded in SIP messages sent to the external realm defined by the SIP NAT and decoded in messages received from the realm.
- **none**—No SIP NAT function is necessary.

The following example shows the SIP home realm configured for a peering network.

```
sip-config
state                           enabled
operation-mode                  dialog
dialog-transparency              disabled
home-realm-id                   acme
egress-realm-id                 -
nat-mode                         public
register-domain                 -
register-host                   -
register-port                   0
initial-timer                   500
max-timer                        4000
trans-expire                     32
invite-expire                    180
inactive-dynamic-conn            32
red-sip-port                     1988
red-max-trans                    10000
red-sync-start-time              5000
red-sync-comp-time               1000
last-modified-date               2005-03-19 12:41:28
```

SIP Interfaces

This section explains how to configure a SIP interface. The SIP interface defines the transport addresses (IP address and port) upon which the Net-Net SBC 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 SBC 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 SBC will use for a SIP interface for the realm. A SIP interface will have one or more SIP ports to define the IP address and port upon which the Net-Net SBC will send and receive messages. For TCP, it defines the address and port upon which the Net-Net SBC will listen for inbound TCP connections for a specific realm.

You need to define at least one SIP port, on which the SIP proxy will listen for connections. If using both UDP and TCP, you must configure more than one port. For example, if a call is sent to the Net-Net SBC using TCP, which it needs to send out as UDP, two SIP ports are needed.

Preferred SIP Port

When a SIP interface contains multiple SIP ports of the same transport protocol, a preferred SIP port for each transport protocol is selected for outgoing requests when the specific SIP port cannot be determined. When forwarding a request that matched a cached registration entry (HNT or normal registration caching), the SIP port upon which the original REGISTER message arrived is used. Otherwise, the preferred SIP port for the selected transport protocol is used. When selecting the preferred SIP port, the default SIP port of 5060 will be selected over other non-default ports.

For SIP interfaces using the SIP NAT function, the preferred SIP port address and port will take precedence over the external address of the SIP NAT when they do not match. If both TCP and UDP SIP ports are defined, the address and port of the preferred UDP port is used.

Proxy Mode

The Net-Net SBC'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 SBC's addresses added. As a result, all subsequent requests are routed through the Net-Net SBC.

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 SBC to perform a global redirect action in response to Redirect messages. Or you can retain the default behavior where the Net-Net SBC sends SIP Redirect responses back to the previous hop (proxy back to the UAC) when the UAS is not a session agent.

The default behavior of the Net-Net SBC 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 SBC can proxy the SIP Redirect response back to the user agent client (UAC) using the value in the session agent's redirect action field (when the UAS is a session agent). If there are too many UASes to define as individual session agents or if the UASs are HNT endpoints, and SIP Redirect responses need to be proxied for UASs that are not session agents; you can set the default behavior at the SIP Interface level.

SIP maddr Resolution

Release S-C6.2.0 provides enhanced resolution of addresses found in SIP contact headers, or in the *maddr* (multicast address) parameter of SIP 3xx REDIRECT messages. Previous releases resolved these addresses as either a host address or as a session agent name. With Release 6.2.0 these addresses can also be resolved as session agent group (SAG) names.

Support for SAG-based resolution is provided by a new **sip-config** parameter, **sag-lookup-on-redirect**. By default, SAG lookup is disabled, providing compatibility with prior releases.

The following sample SIP REDIRECT and ACLI configuration fragment illustrate enhanced processing.

```
Status-Line: SIP/2.0 302 Moved
Message Header
Via: SIP/2.0/UDP
192.168.200.224:5060;branch=z9hG4bKa0fs40009o90sc8oo780.1
From: <sip:1111@192.168.1.222:6000>;tag=1
To: sut <sip:2223@192.168.1.224:5060>;tag=11
Call-ID: 1-28515@192.168.1.222
CSeq: 1 INVITE
Contact: <sip:1111@192.168.1.223;maddr=test.acmepacket.com>
Privacy: user,id;critical;session
P-Preferred-Identity: sipp <sip:sipp@192.168.200.222:5060>
P-Asserted-Identity: abc.com
Subject: abc
Proxy-Require: privacy,prack,abc
Content-Length: 0

session-group
group-name test.acmepacket.com
description
state enabled
app-protocol SIP
strategy Hunt
dest
192.168.200.222
192.168.200.223
...
...
```

In this case, when the 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(config)# session-router
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#

```
2. Use the **sag-lookup-on-redirect** parameter to enable SAG-based *maddr* resolution.
3. Use **done**, **exit**, and **verify-config** to complete SAG-based address resolution.

Trust Mode

The Net-Net SBC supports the Calling Identity privacy requirements based on RFC 3323 and RFC 3325. The trust mode in the SIP interface determines whether the source and destination of a request is a trusted entity. With the implementation of this feature, the Net-Net SBC can understand and support the privacy headers and provide the capability for anonymous packets.

The Net-Net SBC, which acts as a boundary device between the trusted platform and the untrusted Internet, understands the following headers:

- Privacy Header
- P-Asserted-Identity Header
- P-Preferred-Identity Header

Depending on the value of these headers and the mode in which the Net-Net SBC is being operated (B2BUA or the proxy), the appropriate actions are performed.

About the Process

On receiving a message, the Net-Net SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC looks at the value of the Privacy header. If set to *id*, the Net-Net SBC removes all the P-Asserted-Identity headers (if present). It strips the Proxy-Require header if it is set to *privacy*. The Net-Net SBC 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 SBC does not remove the P-Asserted-Identity header fields.

If there is no Privacy header field, the SD will not remove the P-Asserted-Identity headers.

To implement this feature, you need to configure the session agent's trust me parameter to enabled (if the message source is a session agent) and the SIP interface's trust mode to the appropriate value.

Configurable Timers and Counters

SIP timers and counters can be set in the global SIP configuration, and two can be specific for individual SIP interfaces.

You can set the expiration times for SIP messages, and you can set a counter that restricts the number of contacts that the Net-Net SBC tries when it receives a REDIRECT. These are similar to two parameters in the global SIP configuration, trans-expire and invite-expire. You can also set a parameter that defines how many contacts/routes the Net-Net SBC will attempt on redirect.

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. To view all sip-interface parameters, enter a ? at the system prompt.

4. **state**—Enable or disable the SIP interface. The default is **enabled**. The valid values are:
 - enabled | disabled
5. **realm-id**—Enter the name of the realm to which the SIP interface is connected.
6. **sip-ports**—Access the **sip-ports** subelement. See the following section for instructions.
7. **carriers**—Enter the list of carriers related to the SIP interface.
 Entries in this field can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! " \$ % ^ & * () + - = < > ? ' | { } [] @ / \ ' ~ , . _ : ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carrier are valid carrier field formats
8. **proxy-mode**—Enter an option for the proxy mode parameter. Valid values are:
 - **proxy**—Forward all SIP requests to selected targets.
 - **redirect**—Send a SIP 3xx redirect response with the selected target(s) in the Contact header.
 - **record-route**—Forward requests to selected target(s) and insert a Record-Route header with the Net-Net SBC's address. For stateless and transaction mode only.
9. **redirect-action**—Enter the value for the redirect action. Valid values are:
 - **proxy**—Send the SIP request back to the previous hop.
 - **recurse**—Recurses on the Contacts in the response.
 The designated proxy action will apply to SIP 3xx responses received from non-session agents and to 3xx responses received from session agents without configured SIP Redirect message actions (for example, session agents without values for the redirect action field).
10. **contact-mode**—Set the Contact header routing mode, which determines how the contact address from a private network is formatted.
 For example, whether a maddr parameter equal to the Net-Net SBC's SIP proxy needs to be added to a URI present in a Contact header.
 The default is **none**. The valid values are:
 - **none**—The address portion of the header becomes the public address of that private realm.
 - **maddr**—The address portion of the header will be set to the IP address of the Net-Net SBC's B2BUA.
 - **strict**—The contents of the Request-URI is destroyed when a Record-Route header is present.
 - **loose**—The Record-Route header is included in a Request, which means the destination of the request is separated from the set of proxies that need to be visited along the way.
11. **nat-traversal**—Define the type of HNT enabled for SIP. The default is **none**. Valid values are:
 - **none**—HNT function is disabled for SIP.
 - **rport**—SIP HNT function only applies to endpoints that include the rport parameter in the Via header. HNT applies when the sent-by of the topmost

VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address.

- **always**—SIP HNT applies to requests when the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address. (Even when the rport parameter is not present.)
12. **nat-interval**—Set the expiration time in seconds for the Net-Net SBC’s cached registration entry for an HNT endpoint. The default is **30**. The valid range is:
- Minimum—0
 - Maximum—999999999
- Acme Packet recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to cause the UA to send REGISTER messages frequently enough to retain the port binding in the NAT. Retaining the binding lets inbound requests to be sent through the NAT.
13. **tcp-nat-interval**—Set the registration cache expiration time in seconds to use for endpoints behind a NAT device that register using TCP. On upgrade, the Net-Net SBC assigns this parameter the same value as the existing NAT interval. The default is **90**. The valid range is:
- Minimum—0
 - Maximum—999999999
- 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. Requiring endpoints that register over TCP to send refresh requests as frequently as those registering over UDP puts unnecessary load on the Net-Net SBC. By adding a separate configuration for the TCP NAT interval, the load is reduced.
- For upgrade and backward compatibility with Net-Net OS releases prior to Release 4.1, when the `tcpNatInterval` is not present in the XML for a SIP interface configuration, the value of the NAT interval (`natInterval`) is used for the TCP NAT interval as well.
14. **registration-caching**—Enable for use with all UAs, not just those that are behind NATs. The default is **disabled**. The valid values are:
- enabled | disabled
- If enabled, the Net-Net SBC caches the Contact header in the UA’s REGISTER request when it is addressed to one of the following:
- Net-Net SBC
 - registrar domain value
 - registrar host value
- The Net-Net SBC then generates a Contact header with the Net-Net SBC’s address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.
- Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:
- HNT enabled: the Net-Net SBC takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A

cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.

- 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 SBC.

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 SBC, which is the Contact address. Then, the Net-Net SBC forwards the request to the Contact-URI it cached from the original REGISTER message.

15. **min-reg-expire**—Set the time in seconds for the SIP interface. The value you enter here sets the minimum registration expiration time in seconds for HNT registration caching. The default is **300**. The valid range is:

- Minimum—0
- Maximum—999999999

This value defines the minimum expiration value the Net-Net SBC places in each REGISTER message it sends to the real registrar. In HNT, the Net-Net SBC caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the Net-Net SBC to send frequent registrations on to the real registrar.

16. **registration-interval**—Set the Net-Net SBC’s cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the Net-Net SBC to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires. The default is **3600**. The valid range is:

- Minimum—0

A registration interval of zero causes the Net-Net SBC to pass back the expiration time set by and returned in the registration response from the registrar.

- Maximum—999999999

If the expiration time you set is less than the expiration time set by and returned from the real registrar, the Net-Net SBC 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 SBC. You can then define a longer interval time in the registration interval field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

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 SBC’s address. Because the registrar host is the real registrar, it should send the requests back to the Net-Net SBC with the Net-Net SBC’s address in the Request-URI. The default is **disabled**. The valid values are:

- enabled | disabled

For example, you should enable routing to the registrar if your network uses a Net-Net SBC and needs requests to go through its service proxy, which is defined in the registrar host field.

18. **teluri-scheme**—Enable to convert SIP URIs to *tel* (resources identified by telephone numbers) URIs.

If enabled, the requests generated on this SIP interface by the Net-Net SBC will have a *tel* URI scheme instead of the SIP URI scheme. Only the Request, From, and To URIs are changed to the *tel* scheme. After the dialog is established, the URIs are not changed. The default is **disabled**. The valid values are:

- enabled | disabled

19. **uri-fqdn-domain**—Change the host part of the URIs to the FQDN value set here. If set to enabled, and used with an FQDN domain/host, the requests generated by the Net-Net SBC on this SIP interface will have the host part of the URI set to this FQDN value. Only the Request, To, and From URIs are changed. After the dialog is established, the URIs are not changed.

20. **trust-mode**—Set the trust mode for the SIP interface, which is checked by the Net-Net SBC when it receives a message to determine whether the message source is trusted. The default is **all**. Available options are:

- **all**—Trust all SIP elements (sources and destinations) in the realm(s), except untrusted session agents. Untrusted session agents are those that have the **trust-me** parameter set to **disabled**.
- **agents-only**—Trust only trusted session agents. Trusted session agents are those that have the **trust-me** parameter set to **enabled**.
- **realm-prefix**—Trust only trusted session agents, and source and destination IP addresses that match the IP interface's realm (or subrealm) address prefix. Only realms with non-zero address prefixes are considered.
- **registered**—Trust only trusted session agents and registered endpoints. Registered endpoints are those with an entry in the Net-Net SBC's registration cache.
- **none**—Trust nothing.

Session agents must have one or more of the following:

- global realm
- same realm as the SIP interface
- realm that is a subrealm of the SIP interface's realm

21. **trans-expire**—Set the TTL expiration timer in seconds for SIP transactions. This timer controls the following timers specified in RFC 3261:

- Timer B—SIP INVITE transaction timeout
- Timer F—non-INVITE transaction timeout
- Timer H—Wait time for ACK receipt
- Timer TEE—Used to transmit final responses before receiving an ACK

The default is **0**. If you leave this parameter set to the default, then the Net-Net SBC uses the timer value from the global SIP configuration. The valid range is:

- Minimum—0
- Maximum—999999999

22. **invite-expire**—Set the TTL expiration timer in seconds for a SIP client/server transaction after receiving a provisional response.

You set this timer for the client and the sever by configuring it on the SIP interface corresponding to the core or access side.

The default is **0**. If you leave this parameter set to the default, then the Net-Net SBC uses the timer value from the global SIP configuration. The valid range is:

- Minimum—0
- Maximum—99999999

23. **max-redirect-contacts**—Set the maximum number of contacts or routes for the Net-Net SBC to attempt in when it receives a SIP Redirect (3xx Response). The default is **0**. If you leave this parameter set to the default, then the Net-Net SBC will exercise no restrictions on the number of contacts or routes. The valid range is:

- Minimum—0
- Maximum—10

24. **response-map**—Enter the name of the SIP response map configuration that you want to apply to this SIP interfaces for outgoing responses. This parameter is blank by default.

25. **local-response-map**—Enter the name of the SIP response map configuration that you want to apply to this SIP interfaces for locally-generated SIP responses. This parameter is blank by default.

The following two parameters (**method** and **register-response-expires**) enable a SIP registration response mapping feature that allows you to 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.

26. **method**—Enter the name of the received SIP failure response message you want to map to a 200 OK. There is no default for this parameter, and leaving the parameter empty turns off the SIP registration response mapping feature.

27. **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 99999999. 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.

28. **options**—*Optional.*

Configuring SIP Ports**To configure SIP ports:**

1. From sip-interface, type **sip-ports** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(si p-interface)# sip-ports  
ACMEPACKET(si p-port)#[/pre]

```
2. **address**—Enter the IP address of the host associated with the sip-port entry on which to listen. For example:

```
192.168.11.101
```
3. **port**—Enter the port number you want to use for this sip-port. The default is **5060**. The valid range is:
 - Minimum—1025
 - Maximum—65535
4. **transport-protocol**—Indicate the transport protocol you want to associate with the SIP port. The default is **UDP**. The valid values are:
 - **TCP**—Provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with the retransmission of packets when necessary.
 - **UDP**—Provides a simple message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
 - **TLS**—See the Security chapter of this guide for more information about configuring TLS.
5. **allow-anonymous**—Define the allow anonymous criteria for accepting and processing a SIP request from another SIP element.

The anonymous connection mode criteria includes admission control based on whether an endpoint has successfully registered. Requests from an existing SIP dialog are always accepted and processed. The default is **all**.

The following table lists the available options.

- **all**—All requests from any SIP element are allowed.
- **agents-only**—Only requests from configured session agents are allowed. The session agent must fit *one* of the following criteria:
 - Have a global realm.
 - Have the same realm as the SIP interface
 - Be a sub-realm of the SIP interface's realm.

When an agent that is not configured on the system sends an INVITE to a SIP interface, the Net-Net SBC:

- Refuses the connection in the case of TCP.
- Responds with a 403 Forbidden in the case of UDP.
- **realm-prefix**—The source IP address of the request must fall within the realm's address prefix or a SIP interface sub-realm. A sub-realm is a realm that falls within a *realm-group* tree. The sub-realm is a child (or grandchild, and so on) of the SIP interface realm.

Only realms with non-zero address prefixes are considered. Requests from session agents (as described in the **agents-only** option) are also allowed.

- **registered**—Only requests from user agents that have an entry in the registration cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed from any user agent.

The registration cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the **agents-only** option) are also allowed.

- **register-prefix**—Only requests from user agents that have an entry in the Registration Cache (regular or HNT) are allowed; with the exception of a REGISTER request. A REGISTER request is allowed only when the source IP address of the request falls within the realm address-prefix or a SIP interface sub-realm. Only realms with non-zero address prefixes are considered.

The Registration Cache entry is only added if the REGISTER is successful. Requests from configured session agents (as described in the **agents-only** option) are also allowed.

SIP: PRACK Interworking

When you configure your Net-Net SBC with PRACK interworking for SIP, you enable it to interwork between endpoints that support RFC 3262, *Reliability of Provisional Responses in the Session Initiation Protocol*, and those that do not.

As its title indicates, RFC 3262 defines a reliable provisional response extension for SIP INVITEs, which is the 100rel extension tag. While some endpoints do not support the RFC, other SIP implementations require compliance with it. A session setup between two such endpoints fails. However, you can configure your Net-Net SBC to supply the provisional response on behalf of endpoints that do not support it—and thereby enable sessions between those endpoints and the ones requiring RFC 3262 compliance.

How It Works

You need to configure PRACK interworking for a SIP interface associated with the endpoints that need RFC 3262 support. To enable the feature, you set the **100rel-interworking** option. The Net-Net SBC applies PRACK interworking for either the UAC or the UAS. The Net-Net SBC checks to see whether or not it needs to apply PRACK interworking when an INVITE arrives at the ingress or egress SIP interface with the option enabled. First, it checks the Require header for the 100rel tag; if not found there, it checks the Supported header.

Since there is a slight difference in the application of this feature between the UAC and UAS, this section explains both.

UAC-Side PRACK Interworking

The Net-Net SBC applies PRACK interworking on the UAC side when:

- A SIP INVITE contains the 100rel tag in a Require or Supported header
- The ingress SIP interface is enabled with the **100rel-interworking** option
- The UAS fails to send reliable provisional responses

When it is to forward a non-reliable response to a UAC that requires RFC 3262 support, the Net-Net SBC converts the non-reliable response to a reliable one by adding the 100rel tag to the Require header and adding an Rseq header to the response. Further, the Net-Net SBC adds a Require header (complete with the 100rel tag) if there is not one already in the response, and then also adds Rseq header.

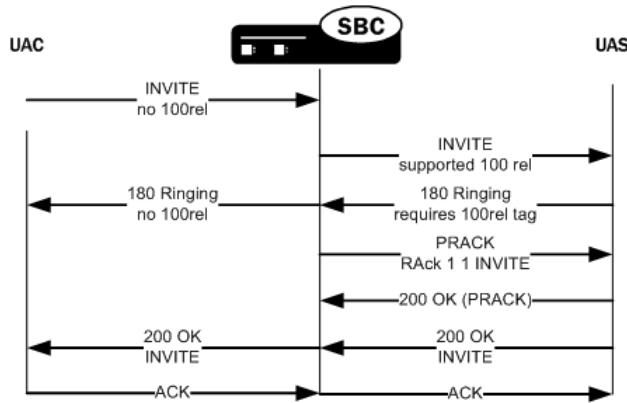
Note that the Net-Net SBC sets the value of the Rseq header as 1 for the first provisional response, and then increments it by 1 for each subsequent provisional

response. It also adds the PRACK method to the Allow header when that header appears.

The Net-Net SBC retransmits the converted reliable provisional response in accordance with RFC 3262, until it receives a PRACK request. For the initial timeout for retransmission, the Net-Net SBC uses the value you set in the **init-timer** parameter in the global SIP configuration. It stops retransmitting when either it receives a transmission, or when the ingress SIP interface's trans-expire timer elapses.

If it never receives a PRACK, the Net-Net SBC does not generate an error response to the INVITE, relying instead on the downstream UAS to produce a final response.

The call flow for this application looks like this:



UAS-Side PRACK Interworking

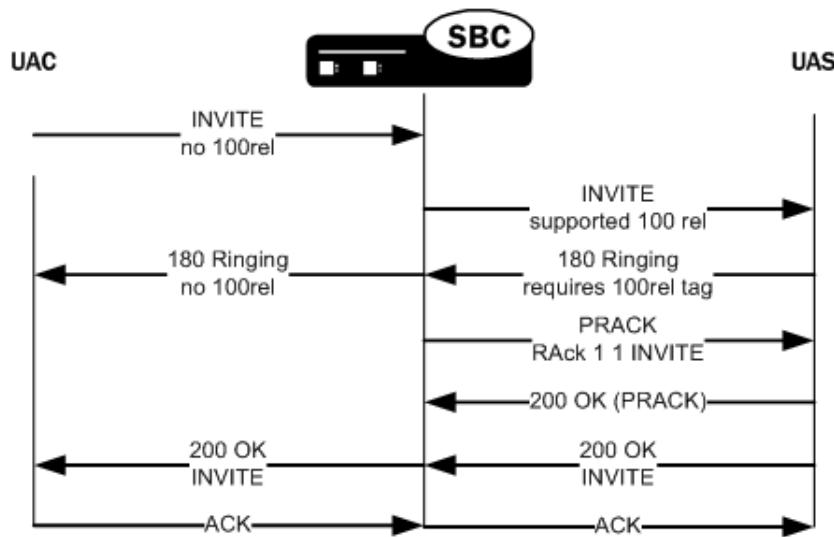
The Net-Net SBC applies PRACK interworking on the UAS side when:

- A SIP INVITE contains the 100rel tag in a Require or Supported header
- The egress SIP interface is enabled with the **100rel-interworking** option
- The UAS does send reliable provisional responses

When the UAC does not support RFC 3262, the Net-Net SBC generates a PRACK request to acknowledge the response. It also converts the response to non-reliable by removing the 100 rel tag from the Require header and removing the RSeq header from the response.

In the case of the UAS, the Net-Net SBC matches the PRACK to a converted reliable provisional response using the PRACK's RAck header. If it finds a matching response, the Net-Net SBC generates a 200 OK to the PRACK. And if it finds no match, then it generates a 481 Call Leg/Transaction Does Not Exist response. The Net-Net SBC generates a 400 Bad Request response if either the RAck is not in the PRACK request or it is not formatted properly.

The call flow for this application looks like this:



ACLI Instructions and Example

You enable PRACK interworking for ingress and egress SIP interfaces. Be sure you know on what side, ingress or egress, you need this feature applied.

To configure PRACK interworking for a SIP interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config**ure **termi**nal
ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **sessi**on-**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)# **si**p-**i**nterface
ACMEPACKET(sip-interface)#
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **100rel-interworking** with a “plus” sign in front of it, and then press <Enter>.
ACMEPACKET(sip-interface)# **opti**ons +**100rel -i**nterworking
If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

Global SIP Timers

This section explains how to configure SIP retransmission and expiration timers.

Note that you can also set timers and counters per SIP interface. For details, refer to the “SIP Interfaces” of this chapter, specifically the [Configurable Timers and Counters \(226\)](#) material.

Overview

SIP timers define the transaction expiration timers, retransmission intervals when UDP is used as a transport, and the lifetime of dynamic TCP connections. The retransmission and expiration timers correspond to the timers defined in RFC 3261.

- init timer: is the initial request retransmission interval. It corresponds to Timer T1 in RFC 3261.

This timer is used when sending requests over UDP. If the response is not received within this interval, the request is retransmitted. The retransmission interval is doubled after each retransmission.

- max timer: is the maximum retransmission interval for non-INVITE requests. It corresponds to Timer T2 in RFC 3261.

The retransmission interval is doubled after each retransmission. If the resulting retransmission interval exceeds the max timer, it is set to the max timer value.

- trans expire: is the transaction expiration timer. This value is used for timers B, D, F, H and J as defined in RFC 3261.
- invite expire: defines the transaction expiration time for an INVITE transaction after a provisional response has been received. This corresponds to timer C in RFC 3261.

If a final response is not received within this time, the INVITE is cancelled. In accordance with RFC 3261, the timer is reset to the invite expire value when any additional provisional responses are received.

- inactive dynamic conn timer defines the idle time (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 value in milliseconds for a response to an INVITE request, and it applies to any SIP request in UDP. In RFC 3261, this value is also referred to as TIMER_T1. The default is **500**. The valid range is:
 - Minimum—0
 - Maximum—999999999
5. **max-timer**—Enter the maximum transmission timeout (T2) for SIP in milliseconds.

When sending SIP over UDP, a re-transmission timer is used. If the timer expires and the message is re-transmitted, the re-transmission timer is then set to twice the previous value (but will not exceed the maximum timer value). Using the default values of 500 milliseconds and 4000 milliseconds, the re-transmission timer is 0.5, then 1, 2, and finally 4. The incrementing continues until the transmission expire timer activates. The default is **4000**. The valid range is:

- Minimum—0
 - Maximum—99999999
6. **trans-expire**—Enter the transaction expire timeout value (Timer B) in seconds to set the time for SIP transactions to live. The same value is used for Timers D, F, H and J. The default is **32**. The valid range is:
- Minimum—0
 - Maximum—99999999
7. **invite-expire**—Enter the invite expire timeout value (Timer C) in seconds to indicate the time for SIP client transaction will live after receiving a provisional response. The default is **180**. The valid range is:
- Minimum—0
 - Maximum—99999999
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. The default value is **32**. The valid range is:
- Minimum—1
 - Maximum—99999999

The following example shows SIP config timer values for a peering network. Some parameters are omitted for brevity.

```
sip-config
      state          enabled
      operation-mode  dialog
      dialog-transparency  dialog
      home-realm-id    acme
      egress-realm-id
      nat-mode         public
      registrar-domain
      registrar-host
      registrar-port   0
      init-timer       500
      max-timer        4000
      trans-expire     32
      invite-expire    180
      inactive-dynamic-conn 32
```

SIP 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 with exceed the bandwidth limitations for parent or child realms and steering pools.

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 `i p`). 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 `i p`). 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.
- `i p`—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 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.

Notes on HA Nodes

This feature has been implemented so that a newly active system is able to perform SIP per user CAC. The standby Net-Net SBC is updated with the appropriate parameters as part of the SIP session update.

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(real-m-config)#
 4. Select the realm where you want to add SIP per user CAC.
ACMEPACKET(real-m-config)# **select**
 5. **user-cac-mode**—Set this parameter to the per user CAC mode that you want to use. The default value is **none**. The valid values are:
 - **none**—No user CAC for users in this realm
 - **aor**—User CAC per AOR
 - **ip**—User CAC per IP
 6. **user-cac-sessions**—Enter the maximum number of sessions per user for dynamic flows to and from the user. The default is 0. Leaving this parameter set to its means that there is unlimited sessions, meaning that the per user CAC feature is disabled in terms of the constraint on sessions. The valid range is:
 7. Minimum—0

8. Maximum—999999999
9. **user-cac-bandwidth**—Enter the maximum bandwidth per user for dynamic flows to and from the user. The default is 0 and leaving this parameter set to the default means that there is unlimited bandwidth, meaning that the per user CAC feature is disabled in terms of the constraint on bandwidth. The valid range is:
 - Minimum—0
 - Maximum—999999999

SIP Per-Realm CAC

Building on the Net-Net SBC's pre-existing call admission control methods, CAC can be performed based on how many minutes are being used by SIP or H.323 calls per-realm for a calendar month.

In the realm configuration, you can now set a value representing the maximum number of minutes to use for SIP and H.323 session using that realm. Although the value you configure is in minutes, the Net-Net SBC performs CAC based on this value to the second. When you use this feature for configurations with nested realms, the parent realm will have the total minutes for all its child realms (i.e., at least the sum of minutes configured for the child realms).

How It Works

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.

Note: The Net-Net SBC does not reject GETS/NSEP calls based on monthly minutes CAC.

You can change the value for minutes-based CAC in a realm configuration at any time, though revising the value downward might cause limits to be reached. This value resets to zero (0) at the beginning of every month, and is checkpointed across both system in an HA node. Because this data changes so rapidly, however, the value will not persist across and HA node if both systems undergo simultaneous failure or reboot.

You can use the ACLI **show monthly minutes <realm-id>** command (where **<realm-id>** is the realm identifier of the specific realm for which you want data) to see how many minutes are configured for a realm, how many of those are still available, and how many calls have been rejected due to exceeding the limit.

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 Realm-Based CAC

Note that setting the new monthly-minutes parameters to zero (0), or leaving it set to its default of 0, disables this feature.

To configure minutes-based CAC:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **config 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(real m-config)#
4. Select the realm where you want to add SIP per user CAC.
 ACMEPACKET(real m-config)# **select**
5. **monthly-minutes**—Enter the number of minutes allowed during a calendar month in this realm for SIP and H.323 calls. By default, this parameter is set to zero (0), which disabled monthly minutes-based CAC. You can enter a value as high as 71582788.
6. Save and activate your configuration.

Viewing Realm-Based CAC Data

Use the ACLI show monthly-minutes command to see the following information:

- How many minutes are configured for a realm
- How many of those are still available
- How many calls have been rejected due to exceeding the limit

To view information about SIP per user CAC using the IP address mode:

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
```

SIP Options Tag Handling

This section explains how to configure SIP options on a global or per-realm level and how to specify whether the feature treatment applies to traffic inbound to or outbound from a realm, or both.

SIP extensions that require specific behavior by UAs or proxies are identified by option tags. Option tags are unique identifiers used to designate new options (for example, extensions) in SIP. These option tags appear in the Require, Proxy-Require, and Supported headers of SIP messages.

Option tags are compatibility mechanisms for extensions and are used in header fields such as Require, Supported, Proxy-Require, and Unsupported in support of SIP.

The option tag itself is a string that is associated with a particular SIP option (i.e., an extension). It identifies this option to SIP endpoints.

Overview

The SIP specification (RFC 3261) requires that the Net-Net SBC B2BUA reject any request that contains a Require header with an option tag the Net-Net SBC does not support. However, many of these extensions operate transparently through the Net-Net SBC's B2BUA. You can configure how SIP defines the Net-Net SBC'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, the you can configure SIP option tag handling to remove the undesired option tag from the Supported header. You can also specify how option tags in Proxy-Require headers are to be treated.

Configuration Overview

You configure the SIP feature element to define option tag names and their treatment by the Net-Net SBC 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 SBC and outbound packets are those which are going from the Net-Net SBC to the realm.

The following tables lists the SIP option tag parameters you need to configure.

Parameter	Description
name	SIP feature tag name
realm	Realm name with which the feature will be associated. To make the feature global, leave the field empty.
support mode inbound	Action for tag in Supported header in an inbound packet.
require mode inbound	Action for tag in Require header in an inbound packet
proxy require mode inbound	Action for tag in Proxy-Require header in an inbound packet
support mode outbound	Action for tag in Supported header in an outbound packet
require mode outbound	Action for tag in Require header in an outbound packet
proxy require mode outbound	Action for tag in Proxy-Require header in an outbound packet

ACLI Instructions and Examples

To configure SIP option tag handling:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# session-router
3. Type **sip-feature** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# sip-feature
ACMEPACKET(sip-feature)#
From this point, you can configure SIP option tags parameters. To view all sip-feature parameters, enter a ? at the system prompt.
4. **name**—Enter a name for the option tag that will appear in the Require, Supported, or Proxy-Require headers of inbound and outbound SIP messages. You must enter a unique value.

Note: Valid option tags are registered with the IANA Protocol Number Assignment Services under Session Initiation Protocol Parameters. Because option tags are not registered until the SIP extension is published as a RFC, there might be implementations based on Internet-Drafts or proprietary implementations that use unregistered option tags.

5. **realm**—Enter the name of the realm with which this option tag will be associated. If you want to apply it globally across realms, leave this parameter blank.
6. **support-mode-inbound**—*Optional.* Indicate the support mode to define how the option tag is treated when encountered in an inbound SIP message's Supported header. The default value is **pass**. Valid values are:

- **pass**—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **strip**—Indicates the tag should not be included in the outgoing message. Use strip if you do not want the extension used.
7. **require-mode-inbound**—*Optional*. Indicate the require mode to define how the option tag is treated when it is encountered in an inbound SIP message's Require header. The default value is **reject**. The valid values are:
- **pass**—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **reject**—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.
8. **require-mode-inbound**—*Optional*. Indicate the require proxy mode to define how the option tag is treated when encountered in an incoming SIP message's Proxy-Require header. The default is **reject**. The valid values are:
- **pass**—Indicates the B2BUA should include the tag in the corresponding outgoing message.
 - **reject**—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.
9. **support-mode-outbound**—*Optional*. Indicate the support mode to define how the option tag is treated when encountered in an outbound SIP message's Supported header. The default value is **pass**. Valid values are:
- **pass**—Indicates the B2BUA should include the tag.
 - **strip**—Indicates the tag should not be included in the outgoing message. Use strip if you do not want the extension used.
10. **require-mode-outbound**—*Optional*. Indicate the require mode to define how the option tag is treated when it is encountered in an outbound SIP message's Require header. The default value is **reject**. Valid values are:
- **pass**—Indicates the B2BUA should include the tag.
 - **reject**—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.
11. **require-mode-outbound**—*Optional*. Indicate the require proxy mode to define how the option tag is treated when encountered in an outgoing SIP message's Proxy-Require header. The default value is **reject**. The valid values are:
- **pass**—Indicates the B2BUA should include the tag.
 - **reject**—Indicates the B2BUA should reject the request with a 420 (Bad Extension) response. The option tag is included in an Unsupported header in the reject response.

The following example shows SIP option tag handling configured for non-default treatment of option tags.

sip-feature	
name	newfeature
real m	peer-1
support-mode-i nbound	Strip
require-mode-i nbound	Reject
proxy-require-mode-i nbound	Pass

support-mode-outbound	Pass
require-mode-outbound	Reject
proxy-require-mode-outbound	Reject
Last-modified-date	2004-12-08 03:55:05

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 SBC.

Option	Description
add-error-to-tag=no	If present (even when set to no), suppresses the addition of an Acme tag on 3xx-6xx responses.
add-prov-to-tag=no	Prevents the Net-Net SBC from adding a tag parameter to the To header (to-tag) to non-100 provisional responses to INVITE requests. Used when a provisional (101-199) response is received from the UAS on a client transaction without a to-tag. By default, the Net-Net SBC 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 SBC. Use for endpoints that do not understand the Expires parameter in the Contact header.
add-ruri-user=<methods>	Causes a userinfo portion to be added to a Request-URI when one is not present. Used to support the OKI phone, which registers a Contact of just an IP-Address but rejects initial INVITEs if the Request_URI does not have a userinfo part. <methods> is a comma-separated list of methods to which the option should apply. If more than one method is listed, the list must be enclosed in quotes. This option only applies to out-of-dialog requests (no tag parameter in the To header). However, if ACK is listed, it will apply to all ACK requests because an ACK is always supposed to have a to-tag.
allow-notify-no-contact	Prevents the Net-Net SBC from rejecting NOTIFYs with a 400 Bad Request response. NOTIFY requests without Contact header are allowed to pass through the Net-Net SBC instead.

Option	Description
call-id-host=<host>	<p>Causes the Net-Net SBC to include a host part (ID@host) in the Call-ID it generates. <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.</p>
contact-endpoint=<param-name>	<p>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.</p>
contact-firewall=<param-name>	<p>Defines a URL parameter to report the NAT between the Net-Net SBC and the real Contact address of an endpoint in a REGISTRAR message forwarded to a registrar when the Net-Net SBC is doing registration caching for NHT. If <param-name> is not specified, the default value firewall is used. This parameter will be added as a URL parameter in the Contact on the REGISTER message. In order for the registration cache to work properly, the softswitch/registrar is expected to include the endpoint parameter in the Request-URI of any SIP request it forwards for the address-of-record.</p>
disable-privacy	<p>Prevents the change of the P-Preferred-Identity to P-Asserted-Identity and lets the P-Preferred-Identity go through unchanged.</p>
drain-sendonly	<p>Causes the Net-Net SBC to examine the SDP attributes and change sendonly mode to sendrecv. This causes the endpoint receiving the SDP to send RTP, which is required for HNT traversal endpoints to work with media servers. The Net-Net SBC sets up the flow so that RTP coming from the endpoint are dropped to prevent the UA that sent the sendonly SDP from receiving packets. See the option video-sbc-session also.</p>
encode-contact=<prefix>	<p>Causes the Net-Net SBC to encode Contact addresses into the userinfo part of the URI. It applies only to Contact address that usually get the maddr parameter. Use when the Net-Net SBC needs requests sent to the URI in the Contact sent instead to the Net-Net SBC. The host part of the URI will have the Net-Net SBC's address. The <prefix> serves as a place between the original userinfo and the encoded address. If a <prefix> is specified, a default of +SD is used. Without this option, the Net-Net SBC adds a maddr parameter.</p>
fix-to-header	<p>For requests that have the Net-Net- SD's address in both the Request-URI and the To-URI, it sets the hostport of the To-URI to a local policy's next hop target on out-of-dialog requests (no to-tag). This is the default IWF behavior, even without this option configured.</p>

Option	Description
forward-reg-callid-change	<p>Addresses the case when an endpoint reboots and performs a third party registration before its old registration expires. During this re-registration, the contact header is the same as it was pre-reregistration. As a consequence of the reboot, the SIP Call-ID changes. In this situation, the Net-Net SBC does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI. To remedy this problem, the Net-Net SBC now keeps track of the Call-ID in its registration cache. A new option in the SIP interface configuration element forces the Net-Net SBC to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.</p>
global-contact	<p>Addresses interoperability in the Dialog and Presence event packages that are used in hosted PBX and IP Centrex offerings. This option enables persistent URLs in the Contact headers inserted into outgoing SIP messages. If this option is not used, URLs placed in the Contact header of outgoing messages are only valid within the context of the dialog to which the message is associated.</p>
ignore-register-service-route-oos	<p>Prohibits a Register message from using a service route if that service route is an out-of-service session agent.</p>
load-limit=<cpu percentage>	<p>Defines the CPU usage percentage at which the Net-Net SBC should start rejecting calls. Default value is 90%.</p>
lp-sa-match=<match strategy>	<p>Changes the ways local policies and session agents match; accounts for realm in matching process. Strategy choices are: all, realm, sub-realm, interface, and network.</p>
max-register-forward=<value>	<p>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.</p>
max-register-refresh=<value>	<p>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 SBC hardware used, but 300 can be used as an initial value.</p>
max-routes=<number of routes>	<p>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 SBC to only try the first, best, route. Setting this option to 0, or omitting it, lets the Net-Net SBC use all of the routes available to it (with the priority scheme for route matching). When you test a policy using the test-policy ACLI command, this option is not recognized and all options that match the criteria are displayed.</p>

Option	Description
max-udp-length=<maximum length>	<p>Setting this option to zero (0) forces sipd to send fragmented UDP packets. Using this option, you override the default value of the maximum UDP datagram size (1500 bytes; sipd requires the use of SIP/TCP at 1300 bytes).</p> <p>You can set the global SIP configuration's max-udp-length=x option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.</p>
media-release=<header-name>[;<header-param>]	<p>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 SBC decode the data to restore the original SDP, which allows the media to flow directly between endpoints in the same network (that is serviced by multiple Net-Net SBCs).</p> <p>The media release information can appear in the following places:</p> <ul style="list-style-type: none"> • 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> <p>Option includes the following:</p> <ul style="list-style-type: none"> • <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. <p>They identify to where the encoded information is passed. If you do not specify a header, P-Media-Release is used.</p>
no-contact-endpoint-port	<p>Enables the Net-Net SBC to add a URL parameter (defined as an argument to the contact-endpoint option) to the Contact headers of REGISTER messages that it forwards to the registrar when it performs registration caching. The value of the contact-endpoint URL parameter is the real address of the endpoint; and if the endpoint is behind a NAT, this includes the IP address and a port number. However, not all network entities can parse that port number, which is included unconditionally. This feature allows you to configure the exclusion of the port number.</p> <p>Despite the fact that you set this parameter in the global SIP configuration, it is applied only to SIP interfaces. However, you can set a contact-endpoint option in the realm configuration, on which this new parameter has no effect.</p>
refer-to-uri-prefix=<prefix>	<p>Defines a prefix to be matched against the userinfo part of Contact headers (config=), of which the Net-Net SBC should create a B2BUA map. This ensures that outgoing messages include the correct userinfo value. This option is used to enable add-on conferencing.</p>

Option	Description
reg-cache-mode=<mode>	<p>Affects how the userinfo part of Contact address is constructed with registration caching. <mode> values are:</p> <ul style="list-style-type: none"> • 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. <p>The from mode is used with softswitches that do not use the cookies used by the Net-Net SBC. It also helps limit the number of bytes in the userinfo; which might create duplicate contacts. For example, if the Net-Net SBC'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 SBC contact of 1234@5.6.7.8.</p>
reg-contact-user-random	<p>Support the SIP random registered-contact feature. Gives the Net-Net SBC the ability to support endpoints that randomly change their contact usernames every time they re-register. Only applicable to operators who need to support the Japan TTC standard JJ-90.22 in specific applications.</p>
	<p>Applies to cases when an endpoint re-registers with a different contact username, but with the same hostname/IP address and the same address of record (AoR). Without this feature enabled, the Net-Net SBC forwards every re-registration to the registrar with the new contact information without it being considered a registration refresh. The Net-Net SBC forwards it to the Registrar using the same sd-contact as the previous registration.</p>
	<p>When you set this option, the Net-Net SBC does treat such a re-registration as a registration refresh when it is received prior to the half-life time for the specific contact. The Net-Net SBC also uses the new contact username for the Request-URI in requests it sends to the UA, and verifies that the UA uses the correct one when that Net-Net SBC is set to allow-anonymous registered mode.</p>
	<p>NOTE: The registration cache mode is set using the option reg-cache-mode, but regardless of how you configure it, the registration cache mode will be set to contact when SIP random registered-contact feature is enabled.</p>
register-grace-timer	<p>Makes the grace time for the SIP Registration configurable. You can configure the grace timer in seconds.</p>
reinvite-trying=[yes]	<p>Causes the Net-Net SBC to send a 100 Trying for re-INVITEs, which is normally suppressed. If you enter the option name but omit the value yes, the option is still active.</p>

Option	Description
reject-interval=<value>	Acts as a multiplier to increase the value presented to the UAC in the Retry-After field. For example, if reject-interval=5 (reject interval is set to 10); at a 90% rejection rate the Net-Net SBC sends "Retry-After: 45". When rejecting calls because of CPU load limiting, the Net-Net SBC adds a "Retry-After" parameter to the error response (typically 503 Service Unavailable). By default the Net-Net SBC sets the Retry-After value to be 1/10th of the current rejection rate.
reject-register=[no refresh]	Allows REGISTER messages through even during load limiting. By default, REGISTER messages are subject to load limiting.
response-for-not-found=<response code>	Change the 404 Not Found generated by the Net-Net SBC to a different response code.
route-register-no-service-route	<p>Controls how a UA is registered. Option can have three values:</p> <ul style="list-style-type: none"> • 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. <p>Addition idle argument ensures 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. Sample syntax: route-register-no-service-route=refresh;idle</p>
sdp-insert-sendrecv	When a call is initiated, the SDP communicates between call offerer and call answerer to determine a route for the media. Devices can be configured to only send media ("a=sendonly"), to only receive media ("a=recvonly"), or to do both ("a=sendrecv"). Some devices, do not disclose this information. With this option configured, when either the offerer or answerer does not disclose its directional attribute, the Net-Net SBC automatically inserts a "sendrecv" direction attribute to the media session.
set-inv-exp-at-100-resp	Set Timer C when a 100 Trying response is received (instead of waiting until 1xx (> 100) is received). If the Net-Net SBC does not receive a 100 Trying response within Timer B, the call should be dropped because there is a problem communicating with the next hop.

Option	Description
strip-domain-suffix-route	Causes sipd to strip any Router headers from the inbound messages coming to the external address of a SIP NAT; if the message contains a FQDN that matches the configured domain suffix for that SIP NAT.
video-sbc-session	<p>Use with drain-sendonly for conference floor support. When configured with drain-sendonly and when the Net-Net SBC receives an SDP, the Net-Net SBC proxies the m=control and its related a= and c= unchanged. Although media streams are allocated for this m line, an actual flow is not set up.</p> <p>SDP received with the following: m=video a=sendonly</p> <p>is sent out as the following: m=video a=sendonly a=X-SBC-Session</p>

SIP Interface Options

The following table lists the SIP interface options supported by the Net-Net SBC.

Option	Description
100rel-interworking	Enables RFC 3262, <i>Reliability of Provisional Responses in the Session Initiation Protocol support</i> .
contact-endpoint=<endpoint name>	The Net-Net SBC inserts the endpoint IP address and port into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value endpoint is used.
contact-firewall=<firewall name>	The Net-Net SBC inserts the firewall IP address and port into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value firewall is used.
contact-vlan=<VLAN/realm name>	The Net-Net SBC inserts the realm and VLAN ID into the Contact headers as messages egress using that SIP interface. The inserted data is the same as the information received in the Request or Response being forwarded. If the endpoint name is not specified, the default value vlan is used.
dropResponse	The Net-Net SBC drops responses by specified status codes. The option value can contain one or more status codes separated by semicolons. Error ranges can also be entered. If any of the response codes matches then a response is not sent. If the dropResponse option is set in both the sip-interface and the session-agent elements, the session-agent setting takes precedence.

Option	Description
max-udp-length=<maximum length>	Sets the largest UDP packers 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. You can set the global SIP configuration's max-udp-length=x option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.
response-for-not-found=<response code>	Change the 404 Not Found generated by the Net-Net SBC to a different response code.
strip-route-headers	Causes the Net-Net SBC to disregard and strip all route headers for requests received on a SIP interface.
upd-fallback	When a request needs to be sent out on the SIP interface for which you have configured this option, the Net-Net SBC first tries to send it over TCP. If the SIP endpoint does not support TCP, however, then the Net-Net SBC falls back to UDP and tries the request again.
via-header-transparency	Enables the Net-Net SBC 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 SBC still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.

SIP Session Agent Options

The following table lists the SIP session agent options supported by the Net-Net SBC.

Option	Description
dropResponse	The Net-Net SBC drops responses by specified status codes. The option value can contain one or more status codes separated by semicolons. Error ranges can also be entered. If any of the response codes matches then a response is not sent. If the dropResponse option is set in both the sip-interface and the session-agent elements, the session-agent setting takes precedence.
trans-timeouts=<value>	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 prevents the session agent from going out of service because of a non-ping transaction timeout.
via-origin=<parameter-name>	Causes a parameter to be included in the top Via header of requests sent to the session agent. The parameter indicates the source IP address of the corresponding request received by the Net-Net SBC. <parameter-name> defines the name of the parameter. If not specified, the default value origin is used.

SIP Realm Options

The following table lists the SIP session agent options supported by the Net-Net SBC.

Option	Description
number-normalization	Applies to the SIP To URI. (Currently the Net-Net SBC supports number normalization on From and To addresses for both inbound and outbound call legs.) Number normalization includes add, delete, and replace string functions that result in consistent number formats. Number normalization occurs on ingress traffic, prior to the generation of accounting records or local policy lookups. (also applies for H.323 to SIP calls.)

ACLI Instructions and Examples

To configure options:

Labels enclosed in <> indicate that a value for the option is to be substituted for the label. For example, "<value>". In order to change a portion of an options field entry, you must re-type the entire field entry.

1. Navigate to the options parameter in the SIP configuration or SIP interface elements.

2. Enter the following:

```
options <Space> <option name>=<value>"
```

For example, if you want to configure the refer-to-uri-prefix option (the add-on conferencing feature):

- 2a. Type **options**, followed by a <Space>.

- 2b. Type **refer-to-uri-prefix**, followed by an equal sign (=).

- 2c. Type the opening quotation mark ("") followed by **conf**, another equal sign and the closing quotation mark.

- 2d. Press <Enter>.

For example:

```
options refer-to-uri-prefix="conf="
```

If the feature value itself is a comma-separated list, it must be enclosed in quotation marks.

Configuring Multiple Options

You can enter a list of options for this field:

1. Type **options** followed by a space.
2. Within quotation marks, enter the feature names and values of the parameters you need. Separate each one with a comma.
3. Close the quotation marks.
4. Press <Enter>.

For example:

```
ACMEPACKET(sip-config)# options "refer-to-uri-prefix="conf=", encode-contact="+SD", add-ruri-user="INVITE, ACK"
```

Adding an Entry

Enter the new entry with a preceding plus (+) sign. For example:

```
options +response-for-not-found
```

This format allows previously configured options field values to remain intact without requiring re-entry of the entire field value.

SIP Security

This section provides an overview of Net-Net SBC's security capability. Net-Net SBC 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 SBC). In addition, Net-Net SBC security lets legitimate users to still place call during attack conditions; protecting the service itself.

Net-Net SBC 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 SBC 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 SBC 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 SBC 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 SBC DoS protection consists of the following strategies:

- Fast path filtering/access control: involves access control for signaling packets destined for the Net-Net SBC host processor as well as media (RTP) packets. The Net-Net SBC accomplishes media filtering using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted list and a deny list for each Net-Net SBC realm (although the actual devices can be dynamically trusted or denied by the Net-Net SBC based on configuration). You do not have to provision every endpoint/device on the Net-Net SBC, 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 SBC 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 SBC's normal call processing; and subsequently not overwhelm systems beyond it. The Net-Net SBC 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 SBC maintains two host paths, one for each class of traffic (trusted and

Configuration Overview

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.

NAT table entries are used to filter out undesired IP addresses (deny list). After the packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager based on the sender's IP address. NAT table entries are used to distinguish signaling packets coming in from different sources for policing purposes.

You can configure deny rules based on the following:

- ingress realm
- source IP address
- transport protocol (TCP/UDP)
- application protocol (SIP, MGCP)

You can configure guaranteed minimum bandwidth for trusted and untrusted signaling paths.

You can configure signaling path policing parameters for individual source addresses. Policing parameters include:

- peak data rate in bits per second
- average data rate in bits per second
- maximum burst size

SIP Unauthorized Endpoint Call Routing

The 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)# **si p-i nterface**
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.

Configuring Security

See the [Security \(901\)](#) 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 SBC 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 SBC does the following:

- routes SIP packets between the Net-Net SBC's SIP proxy (B2BUA) and external networks (or realms), including the translation of IP address and UDP/TCP port information.
- prevents private IP addresses in SIP message URIs from traveling through the untrusted network. SIP NAT either translates the private address to one appropriate for an untrusted address or encrypts the private address into the URI.

Packets arriving on the external address (at port 5060) are forwarded to the Net-Net SBC's SIP proxy with the source address changed to the home address (at port 5060). When the Net-Net SBC's SIP proxy sends packets to the home address (at port 5060), they are forwarded to the external proxy address (and external proxy port), with the source address changed to the external address (at port 5060).

Note: The SIP config's NAT mode parameter works in conjunction with the SIP NAT function configuration. It identifies the type of realm in which the SIP proxy is located (public or private) and affects whether IPv4 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 SBC'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 SBC 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 `si p@123192.169.200.17:5060` will become the public URI of `si p:123-private-eol mhet2chbl 3@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 `PRI VATE-` and a domain suffix value of `pri vate.com`, the private URI of `si p:192.168.200.17:5060` will become the public URI of `si p:PRI VATE-eol mhet2chbl 3.pri vate.com:5060; maddr=172.16.0.15`.

- **trusted**—The SIP proxy is on a trusted network (the address value you entered when you configured the SIP interface's SIP port parameter), and the SIP NAT's external address is the public address of the external realm/network. When the URI contains the home address value, it is translated to the value set for the external proxy address. When the URI contains the SIP proxy's address, it is translated to the external address. If the URI contains any other private address (matching the realm's address prefix, identified in the SIP NAT's realm ID), the private address is encrypted and the address is replaced with the external address.

Note: Do not use the home proxy address value with private NAT functioning.

Adding a `maddr` Parameter to a URI

When you configure a SIP interface, you can configure the contact mode. The contact mode sets the contact header routing mode, which determines how the contact address from a trusted network is formatted. You set the contact mode to add a `maddr` parameter equal to the SIP proxy to the URI in the Contact header. For example, the URI from the prior example (`si p:192.168.200.17:5060`) becomes `si p:123-trusted-eol mhet2chbl 3@172.16.0.15; maddr=172.16.0.12`.

Note: For SIP elements that do not support the `maddr` parameter, configure a Contact mode as `none`.

You might require this encryption to cause other SIP elements in the untrusted network to send requests directly to the SIP proxy. Otherwise, the requests are sent to the home address. However, responses sent by the SIP proxy will have the SIP proxy's source address, rather than the home address. Some SIP elements might drop responses that come from a IP address different from the one to which the request is sent.

About Headers

You can specify which SIP headers you want effected by the SIP NAT function. The URIs in these headers are translated and encrypted, the encryption occurs according to the rules of this SIP NAT function.

You can enter header values by using either the full header name or its corresponding abbreviation, if applicable. The following table lists the available headers and their corresponding abbreviations

Header	Abbreviation
Call-ID	i
Contact	m
From	f
Record-Route	none
Route	none
Ready-To	none
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 SBC. 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 SBC's IP address in the sent-by portion. If a Contact header is present, it is replaced with one that has the Net-Net SBC'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 SBC maps FQDNs that appear in the certain headers of incoming SIP messages to the IP address that the Net-Net SBC 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 SBC encodes various information in the outgoing message, which is passed from one side to another in SIP transactions.

SIP NAT function cookies let the Net-Net SBC hide headers, IPv4 addresses, and SIP URIs. These cookies are included when certain conditions are present in Net-Net SBC 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 SBC places a cookie in the userinfo portion of a SIP URI when a SIP header contains a SIP URL, 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-pfi 1s7n2pstna@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-pfi 1s7n2pstna. acme. com

URL Parameter

If the SIP NAT function's use url parameter field has a value of *from-to* or *all*, the SIP NAT function places all cookies generated to hide SIP URIs in a custom tag appended to the header. Setting the use url parameter field to:

- `from-to` only affects the behavior of the SIP NAT function's cookies in the `From` and `To` headers.
- `all` affects all SIP headers processed by the SIP NAT function

The cookie is the following:

```
[; url -parameter]=[host nat tag][encrypted 13-byte host IP][encrypted 13-byte maddr IP]
```

where:

- `[; url -parameter]` refers to the SIP NAT function's parameter name field. This cookie type is associated with the `all` and `from-to` field value options of the SIP NAT function's `use url` parameter field.
- `[host nat tag]` refers to the SIP NAT function's host NAT tag field.
- `[encrypted 13-byte host IP]` refers to the host IP encryption.
- `[encrypted 13 byte maddr IP (if present)]` refers to the `maddr` IP encryption, if it exists.

With a host NAT tag of ACME- and a parameter name of `acme_param`, the following SIP-URI:

```
sip:6175551212@192.168.1.100
```

might be translated into the following:

```
sip:6175551212@172.16.1.10;acme_param=ACME-pfi1s7n2pstna.
```

tel URL

The SIP NAT function cookie is used when devices in your network are strict about the context portion of SIP messages regarding the conversion of tel URLs. This cookie for the tel URL parameter portion of a SIP message is the following:

```
"tel " URL parameter-[13-byte host IP][13 byte optional maddr IP]domain suffix
```

where:

- `tel URL parameter` refers to the SIP NAT function's `use url` parameter. This cookie type is associated with the `use url` parameter's `phone` field value for the SIP NAT.
- `[13-byte host IP]` refers to the host IP encryption.
- `[13 byte optional maddr IP]` refers to the `maddr` IP encryption, if it exists.
- `domain suffix` refers to the SIP NAT function's domain suffix field.

Configuration Overview

Configuring the SIP NAT function falls into two areas, the SIP NAT interface parameters and the SIP NAT policies.

SIP NAT Interface

The following tables lists the SIP NAT function interface parameters you need to configure.

Parameter	Description
realm ID	Name of the external realm. The realm ID must be unique; no two SIP NATs can have the same realm ID. This realm ID must also correspond to a valid realm identifier entered when you configured the realm.
external proxy address	IPv4 address of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net SBC communicates. Entries must follow the IP address format.
external proxy port	UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net SBC communicates. Minimum value is 1025, and maximum value is 65535. Default is 5060.
external address	IPv4 address on the media interface in the external realm. Enter a value that ensures any packet with an external address value as its destination address is routed to the Net-Net SBC through the media interface connected to or routable from the external realm. Entries must follow the IP address format. To specify whether the external realm referenced in this field is private or public, configure the SIP config's NAT mode.
home address	IPv4 address on the media interface in the home realm. Enter a value that ensures any packet with a home address value as its destination address must be routed to the Net-Net SBC through the media interface connected to or routable from the home realm. Entries must follow the IP address format. The value entered in this field must be different from the IP address value of the home realm's network interface element. The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.
home proxy address	Sets the IP address for the home proxy (from the perspective of the external realm). By default, this field is empty. An empty home proxy address field value signifies that there is no home proxy, and the external address will translate to the address of the Net-Net SBC's SIP proxy. Entries must follow the IP address format.

Parameter	Description
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. <ul style="list-style-type: none"> • enabled adds route if Request-URI is not the Net-Net SBC • disabled does not route inbound requests to the home proxy • forced always adds route

SIP NAT Function Policies

The following tables lists the SIP NAT function policy parameters you need to configure.

Parameter	Description
domain suffix	Domain name suffix of the external realm. The domain name suffix refers to and must conform to the hostname part of a URI. In combination with the user NAT tag and host NAT tag values, this value is used to help the Net-Net SBC identify an encoded URI that it needs to translate when moving between public and private realms. This suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is ACME-abc123 and the domain-suffix value is .netnetsystem.com, the resulting FQDN will be ACME-abc123.netnetsystem.com.
address prefix	Defines which IPv4 address prefixes from incoming messages require SIP-NAT encoding (regardless of the realm from which these messages came).
tunnel redirect	Controls whether Contact headers in a 3xx Response message received by the Net-Net SBC are NATed when sent to the initiator of the SIP INVITE message.
use url parameter	Establishes whether SIP headers will use the URL parameter entered in the parameter name for encoded addresses that the SIP NAT function creates. Also, if SIP headers will be used, which type of headers will use the URL parameter. For example, all headers or just the From and To headers. Enumeration field.
parameter name	Indicates the name of the URL parameter when use url applies. This field value will be used in SIP NAT encoding addresses that have a use url parameter value of either from-to or all.
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 SBC identify an encoded URI that it needs to translate when moving between public and private realms.

Parameter	Description
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 SBC identify an encoded URI that it needs to translate when moving between public and private realms.
headers	Lists the SIP headers to be affected by the Net-Net SBC's SIP NAT function. The URLs 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:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# **session-router**
3. Type **sip-nat** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-nat**
ACMEPACKET(sip-nat)#
 4. **realm-ID**—Enter the name of the realm you want to identify as the external realm.
The name you use as the realm ID must be unique. No two SIP NAT functions can have the same realm ID. Also, this value must correspond to a valid identifier entry already configured for the realm.
 5. **domain-suffix**—Enter the domain suffix to identify the domain name suffix of the external realm. The domain suffix must begin with a (.) dot.
The domain name suffix refers to and must conform to the hostname part of a URI. For example:
.netnetsystem.com
The domain suffix is appended to encoded hostnames that the SIP NAT function creates. For example, if the encoded hostname is ACME-abc123, the resulting FQDN is ACME-abc123.netnetsystem.com.
 6. **external-proxy-address**—Enter the external proxy address to identify the IPv4 address of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net SBC communicates.
Enter the value in the IP address format. For example:
192. 168. 11. 200
 7. **external-proxy-port**—Enter the external proxy port value to identify the UDP/TCP port of the SIP element (for example, a SIP proxy) in the external network with which the Net-Net SBC communicates. The default is **5060**. The valid range is:
 - Minimum—1025
 - Maximum—65535

8. **external-address**—Enter the external address, which is an IPv4 address on the media interface in the external realm.

Enter the value in the IP address format. For example:

192. 168. 11. 101

This value must be such that any packet with an external address value as its destination address is routed to the Net-Net SBC through the media interface connected to or routable from the external realm.

9. **home-address**—Enter the home address, which is an IPv4 address on the network interface in the home realm. This value must be such that any packet with a home address value as its destination address must be routed to the Net-Net SBC through the media interface connected to or routable from the home realm.

Enter the value in the IP address format. For example:

127. 0. 0. 10

The value entered in this field **must be different** from the IP address value of the home realm's network interface element.

The home realm network interface is associated with this SIP NAT by its realm ID and the realm's identifier and network interface value you entered when you configured the realm. The realm's network interface identifier value corresponds to this SIP NAT's realm ID, the SIP config's home realm ID, and the media manager's home realm ID.

10. **home-proxy-address**—Enter the home proxy address to set the IP address for the home proxy (from the perspective of the external realm).

By default, this field is empty. No home proxy address entry signifies there is no home proxy, and the external address will translate to the address of the Net-Net SBC's SIP proxy.

Enter the value in the IP address format. For example:

127. 1. 0. 10

11. **home-proxy-port**—Enter the home proxy port to set the port number for the home realm proxy. The default value is 0. The valid range is:

- Minimum—0, 1025
- Maximum—65535

12. **route-home-proxy**—*Optional*. Enable or disable requests being routed from a given SIP-NAT to the home proxy. The default value is **disabled**. The valid values are:

- **enabled**—All inbound requests for a specific SIP NAT are routed to the home proxy
- **disabled**—All inbound requests are not routed through the home proxy.
- **forced**—The Request is forwarded to the home proxy without using a local policy.

13. **address-prefix**—*Optional*. Indicate the IPv4 address prefix from incoming messages that requires SIP NAT function encoding (regardless of the realm from which these messages came).

Note: This value overrides the value set in the realm's address prefix field.

This field's format incorporates an IPv4 address and number of bits in the network portion of the address. For example, a Class C address has a 24-bit network part. The address prefix for 101.102.103.x would be represented as 10.102.103.0/24.

The default value is an asterisk (*). When you enter this value or do not enter a value, the realm's address prefix value is used.

14. **tunnel-redirect**—Set to one of the following values to indicate whether certain headers in a 3xx Response message received by the Net-Net SBC are NATed when sent to the initiator of the SIP INVITE message. The default is **disabled**. The valid values are:

- **enabled**—Certain headers in a 3xx Response message are NATed.
- **disabled**—Certain headers in a 3xx Response message are not NATed.

15. **use-url-parameter**—Establish whether SIP headers will use the URL parameter (configured in the next step) for encoded addresses created by the SIP NAT function. If SIP headers will be used, this value identifies which types of headers will use the URL parameter. The default value is **none**. The available values include:

- **none**—No headers will use the URL parameter for address encoding.

The following example illustrates the functionality of a Net-Net SBC using a use url parameter value of none:

`sip: 1234@1.2.3.4` is translated into `sip: 1234-acme-xxxx@5.6.7.8`

where -acme-xxxx is a cookie and xxxx is the encoded version of 1. 2. 3. 4.

- **from-to**—From and To headers will use the URL parameter for address encoding

The following example illustrates the functionality of a Net-Net SBC using a use url parameter value of none:

`sip: 1234@1.2.3.4` is translated into `sip: 1234@5.6.7.8; pn=acme-xxxx`

where -acme-xxxx is a cookie and xxxx is the encoded version of 1. 2. 3. 4.

- **all**—All headers will use the URL parameter for address encoding. Acme Packet recommends not using this values because other SIP elements or implementations (other than the Net-Net SBC) might not retain the URL parameter in subsequent SIP messages that they send to the Net-Net SBC.
- **phone**—

If this field is set to either **from-to** or **all**, the Net-Net SBC puts the encoded address of the SIP NAT into a URL parameter instead of using the encoding name inside the userinfo part of the address.

16. **parameter-name**—If you have configured the **use-url-parameter** with the from-to or all value, you need to indicate the hostname prefix.

The parameter name value is used in SIP NAT encoding addresses that have the use url parameter values of from-to or all.

17. **user-NAT-tag**—Enter a value to identify the username prefix used for SIP URIs. The values you can use can include any characters valid for the userinfo part of a URI. This should be made unique for each realm and SIP NAT function.

The default value is **-acme-**.

In combination with the domain suffix and host NAT tag values, this value is used to help the Net-Net SBC identify an encoded URI that it needs to translate when moving between public and private realms.

18. **host-NAT-tag**—Enter a value for the host NAT tag field to identify the hostname prefix used for SIP URIs. The value refers to domain labels and can include any characters valid for the hostname part of the URI. This should be made unique for each realm and SIP NAT function.

The default value is ACME-.

In combination with the domain suffix and user NAT tag values, this value is used to help the Net-Net SBC identify an encoded URI that it needs to translate when moving between public and private realms.

19. **headers**—List the SIP headers you want affected by the SIP NAT function. The URIs in these headers are translated and encrypted, and encryption occurs according to the SIP NAT function rules.

To enter the full default list, type **headers**, followed by a <Space> and -d, then press <Enter>.

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-i p-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 SBC's external address

To view all SIP NAT function parameters, enter a ? at the system prompt. The following example shows SIP NAT configuration for peering network.

```
si p-nat
      real m-i d          peer-1
      domai n-suffi x     . 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 -redi rect  disabled
      use-url -parameter none
      parameter-name      -p1-
      user-nat-tag        P1-
      host-nat-tag

      headers             Call-ID Contact From Join Record-Route
                          Refer-To Replaces Reply-To Route To Via
                          f i m r t v
```

SIP Realm Bridging

This section explains how to configure the internal routing among realms known as realm bridging. Realm bridging lets you cross-connect SIP interfaces. You can use one of the following two methods for bridging realms:

- local policy bridging: use this method to enable dynamic internal routing between realms if your SIP interfaces do not have the SIP NAT function applied.
- SIP NAT bridging: use this method if your SIP interfaces have the SIP NAT function applied.

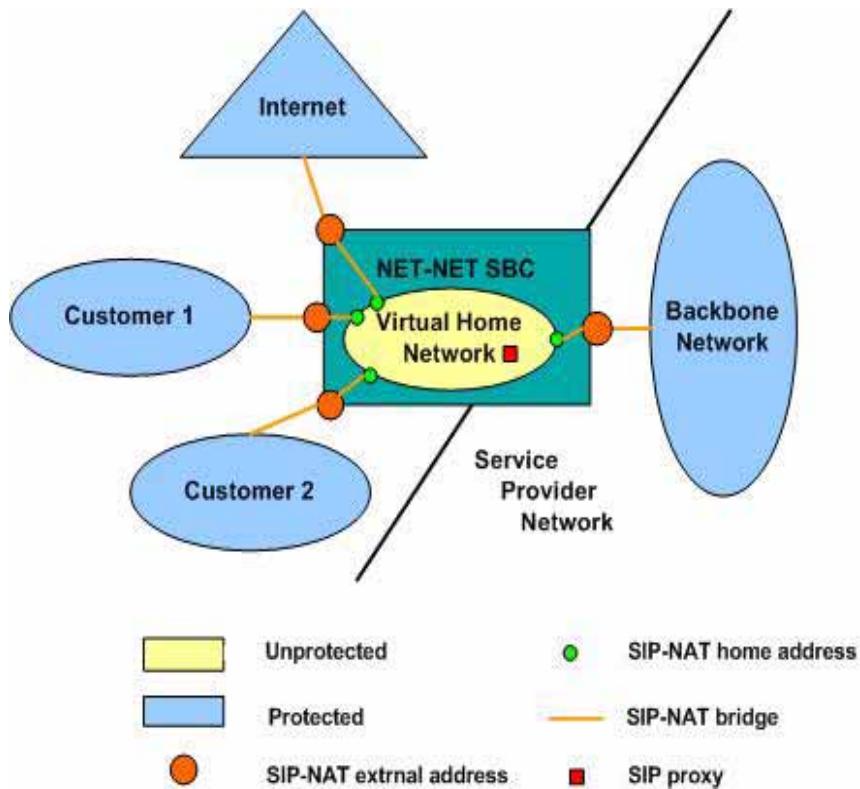
About SIP NAT Bridging

Each SIP NAT has a presence in two realms, trusted and untrusted. The SIP NAT bridge is the conduit for packages in and out of the home realm. It creates a bridge between realms by providing address translations; removing all references to the original IP addressing from the packets sent to the destination network.

With the SIP NAT bridge, an untrusted (or public) home network can reside within the Net-Net SBC, 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 SBC's SIP proxy (the SIP server that receives SIP requests and forwards them on behalf of the requestor).

The following diagram illustrates this unprotected (or public) and protected (or private) division.



SIP NAT Bridge Configuration Scenarios

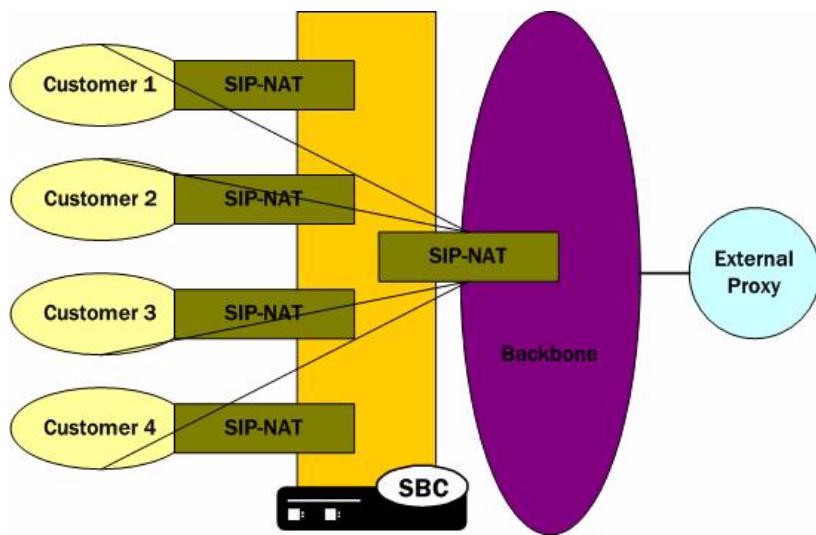
You can configure the SIP NAT bridge functionality in a many-to-one or a one-to-one relationship. For example, multiple customer SIP NATs can be tied to a single backbone SIP NAT, or a single customer SIP NAT can be tied to a single backbone SIP NAT.

You might need to use several SIP NATs on the customer side while using only one on the backbone side in a many-to-one relationship. Or you might configure one SIP NAT on the backbone side for every one that you configure on the customer side in a one-to-one relationship.

You can route all customer side SIP NAT requests to the corresponding backbone SIP NAT regardless of the Request URI. If a request arrives from the customer network with a Request URI that does not match the customer SIP NAT external address or the local policy that would route it to the backbone SIP NAT; the route home proxy value is used.

Many to One Configuration

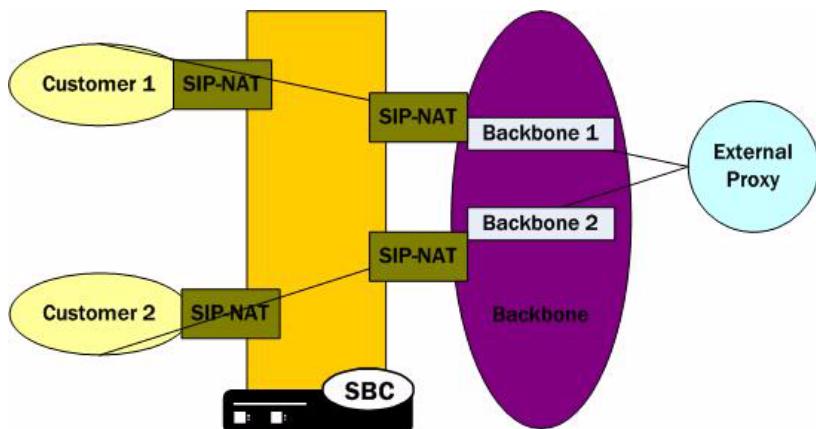
In the many-to-one scenario, multiple customer SIP NATs are tied to a single backbone SIP NAT. The following diagram illustrates the many-to-one SIP NAT bridge configuration.



One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. On the backbone SIP NAT side, you configure the home proxy address to match the home address of the customer SIP NAT. On the customer side, you configure the home proxy address to match the home address of the backbone SIP NAT.

The following diagram illustrates the one-to-one SIP-NAT bridge configuration.



SIP NAT Bridge Configuration

You create a bridge between SIP NATs by pointing them at one another. You point the SIP NATs at each other by configuring the home address and home proxy address to create the bridge. In addition, you can configure the route home proxy on the customer's side of a SIP NAT to force all requests to be routed to the corresponding backbone SIP NAT, regardless of the Request URI. You need to force requests when elements in the customer's network send requests with a Request URI that does not match the customer's SIP NAT external address. Or when the

Request URI does not match a local policy element that would route the requests to the backbone SIP NAT.

You also need a home network to create a SIP NAT bridge. If you do not have a real home network, you need to create a virtual one. You also need to configure instances of the SIP NAT to create the SIP NAT bridge within your network.

Creating a Virtual Home Network

A virtual home network is a home network that resides entirely within the Net-Net SBC, 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 SBC.

The internal home realm/network is usually configured with addresses within the special loopback range (127.0.0.0/8) as described in RFC 3330. This applies to the SIP port addresses for the home realm's SIP interface, and all home addresses for SIP NATs. The address 127.0.0.1 should not be used because it conflicts with the default loopback interface setup by the system for inter-process communication.

To create a virtual home network:

1. Set the name and subport ID of the network interface associated with the home realm element to lo0:0.
2. To enable the SIP proxy to listen for messages on the virtual home realm, configure the home realm ID. It must correspond to the realm's identifier, in which you set the network interface subelement to point to the appropriate network interface element.

The following table lists the field values you need to set when you are using SIP NAT bridge functionality and you do not have a real home network.

Configuration Element	Sample Values	
realm configuration	identifier	home
	network interfaces	lo0:0
	address prefix	127.0.0.0/8
SIP configuration	home realm ID	home
	SIP ports address	127.0.0.100

Many-to-One Configuration

To configure many-to-one:

1. For the backbone SIP NAT, ensure the home proxy address field is blank.
2. For the customer side SIP NAT:
 - 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.
 - 2c. Set route home proxy to forced.

The following table lists the field values you need to set to create a many-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IPv4 address of the home realm. For example: 127.0.0.120
	home proxy address	IPv4 address of the home proxy from the perspective of the external realm. For a backbone SIP NAT, leave blank.
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110
	route home proxy	forced

One-to-One Configuration

In the one-to-one scenario, a single customer SIP NAT is tied to a single backbone SIP NAT. The home proxy address field value of the backbone SIP NAT must match the home address of the customer SIP NAT. On the customer side, the home address of the customer SIP NAT should be defined as the home address of the customer, the home proxy address field value should match the home address of the backbone SIP NAT, and route home proxy should be set to forced.

The following table lists the field values you need to set to create a one-to-one SIP NAT bridge.

SIP NAT Entity	Field	Sample Values
Backbone SIP NAT	home address	IPv4 address of the home realm. For example: 127.0.0.110
	home proxy address	IPv4 address of the home proxy from the perspective of the external realm. 127.0.0.120
Customer SIP NAT	home address	127.0.0.120
	home proxy address	127.0.0.110
	route home proxy	forced

Shared Session Agent

Usually, the same set of servers (the external proxy) is used for all SIP NATs to the backbone network. In order to support redundant servers in the backbone of a SIP NAT bridge, the original egress realm as determined by the incoming Request URI needs to be retained after a local policy lookup.

When a request arrives at the Net-Net SBC, 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 SBC retains the original session agent's realm ID. Because the target session agent does not have a realm ID defined, the original egress realm is retained.

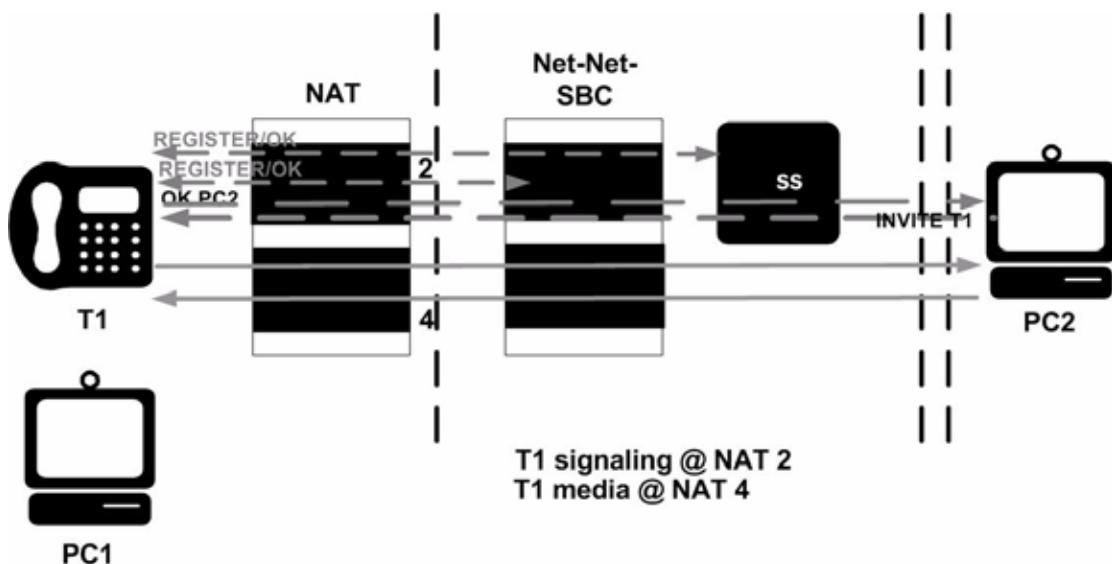
SIP Hosted NAT Traversal (HNT)

This section explains how to configure SIP Hosted Network Address Translation (HNT) traversal. SIP HNT lets endpoints behind a NAT/firewall device send and receive signaling and media using the Net-Net SBC as a relay.

About SIP HNT

SIP HNT is a technique the Net-Net SBC uses to provide persistent reachability for SIP UAs located in private Local Area Networks (LANs) behind Nat/firewall devices. It relies on frequent, persistent messaging to ensure that the binding on the intermediary NAT device is not torn down because of inactivity. HNT does not require support for the NAT in the SIP endpoint.

The following diagram illustrates SIP HNT traversal.



The Net-Net SBC's HNT function allows endpoints located behind NATs to communicate; providing means to traverse NATs. The Net-Net SBC interacts with endpoints (using SIP) to allow persistent inbound and outbound signaling and media communications through these NATs.

The Net-Net SBC automatically detects when an intermediate NAT exists between the UA and the Net-Net SBC by comparing the Layer 3 IP address of a REGISTER message with the IP address indicated within the UA. The Net-Net SBC sends signaling responses to the address and port that the request came from, rather than the address and port indicated in the request. The Via header in the request message indicates where the response should be sent.

Using HNT with Existing NAT Device

For network architectures in which premise devices and endpoints reside behind an existing NAT device, the Net-Net SBC'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 SBC uses periodic endpoint registration messages to dynamically establish and maintain bindings in the NAT. These bindings keep a signaling port (port that is opened on a firewall to allow traffic to pass through it is a pinhole) open in the NAT that allows the inbound signaled communications to pass through. Using the endpoint registrations, the Net-Net SBC then maps the Layer 3 (OSI network layer that deals with switching and routing technologies for data transmission between network devices) IPv4 address/port information from the NAT device to the Layer 5 (OSI session layer that deals with session and connection coordination between applications) entity (for example, user name or phone number) behind the NAT so that when an incoming signaling message is received, the Net-Net SBC 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 SBC, it can identify the IPv4 address/port information on the NAT device used for the outgoing media coming from the user name/phone number. The Net-Net SBC then uses that same NAT's IPv4 address/port information to send incoming media to the correct user name/phone number behind the NAT device.

Prerequisites

In order to achieve HNT, the endpoints involved must be capable of:

- symmetric signaling: sending and receiving SIP messages from the same transport address (IP address or User Datagram Protocol/Transmission Control Protocol (UDP/TCP) port
- symmetric media: sending and receiving Real-Time Transport Protocol (RTP) messages from the same UDP port

These conditions are required to allow signaling and media packets back through the NAT (through the bound external address and port). These packets must come from the address and port to which the outbound packet that created the NAT binding was sent. The NAT sends these inbound packets to the source address and port of the original outbound packet.

When SIP HNT is used, the Net-Net SBC sends signaling responses to the address and port that the request came from rather than the address and port indicated in the request. The Via header in the request message indicates where the response should be sent.

Keeping the NAT Binding Open

Additional measures are also required to keep the NAT binding open because most NAT bindings are discarded after approximately a minute of inactivity. The Net-Net SBC 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 SBC maintains the registration state. When an endpoint first registers, the Net-Net SBC 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.

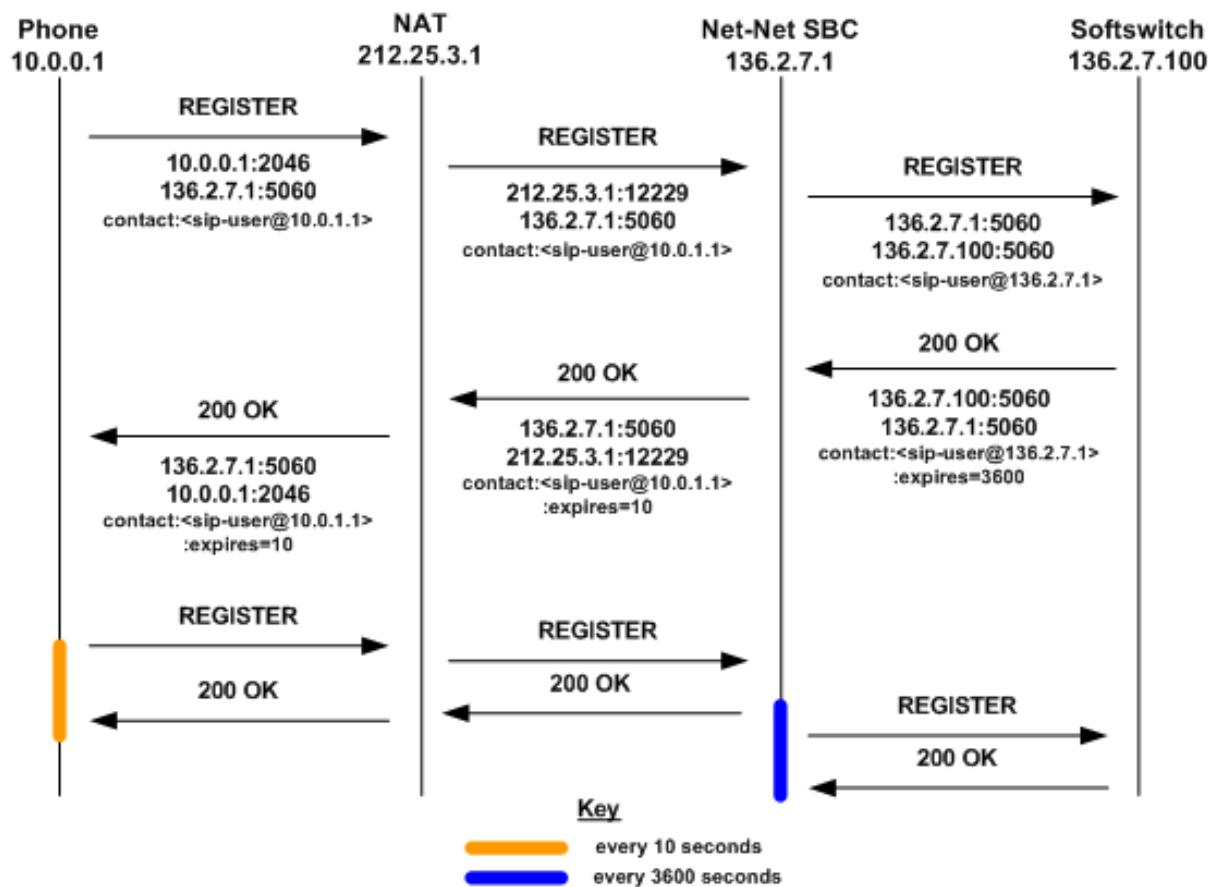
Note: A registrar can be located in a SIP NAT realm.

When a successful response is received, the Net-Net SBC caches the registration to memory. This cached registration lives for the length of time indicated by the expiration period defined in the REGISTER response message from the registrar. The response sent back to the endpoint has a shorter expiration time (defined by the SIP config's NAT interval) that causes the endpoint to send another REGISTER message within that interval. If the endpoint sends another REGISTER message before the cached registration expires, the Net-Net SBC responds directly to the endpoint. It does not forward the message to the real registrar.

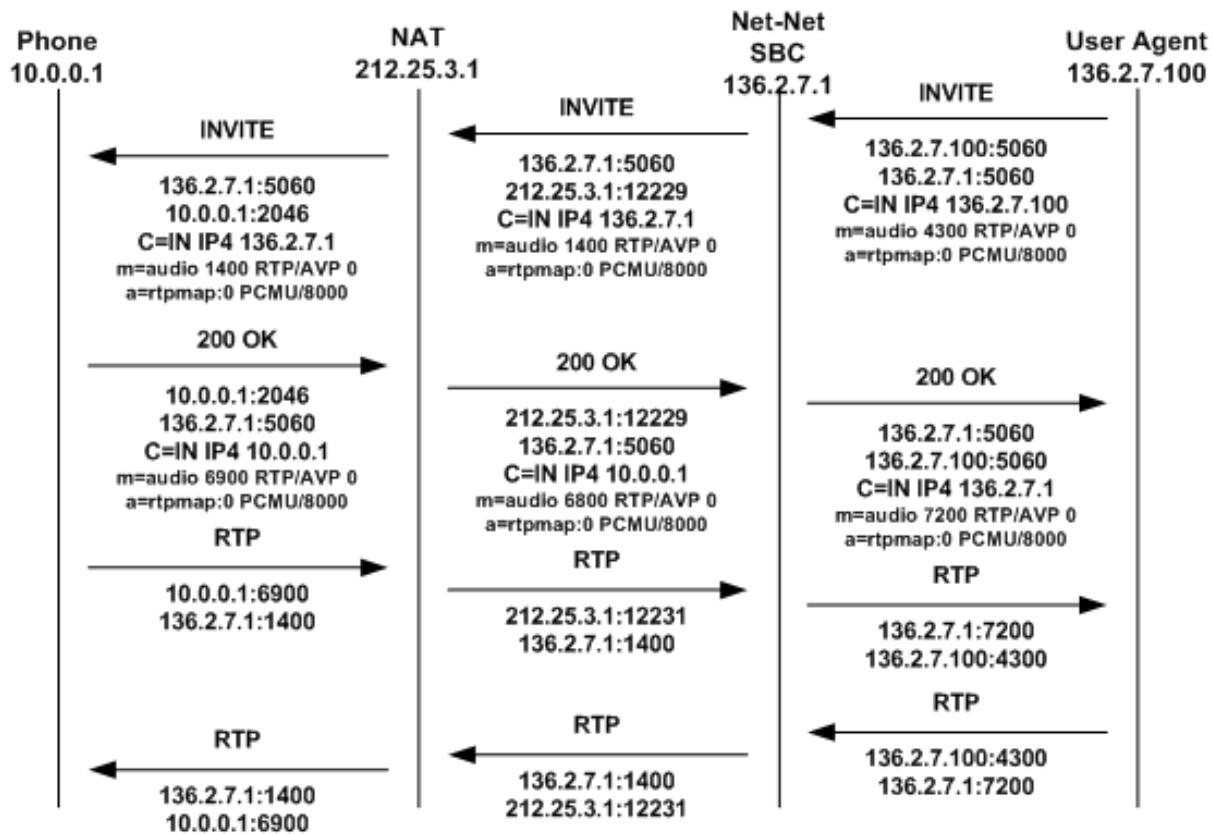
If the cached registration expires within the length of time indicated by the NAT interval, the REGISTER message is forwarded to the real registrar. If the Net-Net SBC does not receive another REGISTER message from the endpoint within the length of time indicated by the NAT interval, it discards the cached registration.

The Contact Uniform Resource Identifier (URI) in the REGISTER message sent to the registrar by the Net-Net SBC points at the Net-Net SBC so that the proxy associated with the real registrar sends inbound requests to the Net-Net SBC. 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 SBC 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 you enter an asterisk (*) for the register host. An FQDN consists of an unlimited number of domain labels (domain names), each separated by a dot (.). The FQDN can include the top level domain name (for example, acmepacket.com).

In the hostname acme-packet.domainlbl.example100.com, the syntax is as follows:

- acme-packet is a domain label
- domainlbl is a domain label
- example100 is a domain label
- com is the top label

The information configured in a local policy is used before DNS is used. If the next hop destination address (defined in the local policy's next hop field) is an IPv4 address, a DNS server is not needed. A DNS server is needed when the IPv4 address

of the next hop destination address is a FQDN or cannot be determined from the Net-Net SBC'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 SBC always uses the configured address. You can limit the number of endpoints that receive the HNT function. For example, you can use a non-wildcarded registrar domain field value (like acme.com) with a wildcarded registrar host field value.

HNT Configuration Overview

SIP HNT Single Domain Example

To configure SIP HNT NAT traversal, you need to configure both the SIP interface and the SIP config.

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a single domain and registrar.

- SIP config

Parameter	Sample Value
registrar domain	netnetsystem.com
registrar host	192.168.12.1
registrar port	5060

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60
minimum registration expire	200
registration caching	disabled
route to registrar	enabled

SIP HNT Multiple Domain Example

The following example shows values entered for the SIP config and SIP interface elements to configure SIP HNT for a multiple domains and multiple registrars.

- SIP config

Parameter	Sample Value
registrar domain	*
registrar host	*
registrar port	0

- SIP interface

Parameter	Sample Value
NAT traversal	always
NAT interval	60

Parameter	Sample Value
minimum registration expire	200
registration caching	disabled
route to registrar	enabled

ACLI Instructions and Examples

To configure a SIP interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# session-router
3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
From this point, you can configure physical interface parameters. To view all SIP interface parameters, enter a ? at the system prompt.
4. **nat-traversal**—Define the type of HNT enabled for SIP. The default value is **none**. Available values include:
 - **none**—Disables the HNT feature for SIP (default value)
 - **rport**—SIP HNT function only applies to endpoints that include the rport parameter in the Via header and the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address.
 - **always**—SIP HNT applies to requests when the sent-by of the topmost VIA matches the Contact-URI host address, both of which must be different from the received Layer 3 address. (Even when the rport parameter is not present.)
5. **nat-interval**—Set the expiration time in seconds for the Net-Net SBC's cached registration entry for an HNT endpoint. The default value is **30**. The valid range is:
 - Minimum—0
 - Maximum—999999999

Acme Packet recommends setting the NAT interval to one-third of the NAT binding lifetime. A NAT binding lifetime is the network connection inactivity timeout. The value is configured (or hardwired) in the NAT device (firewall). This timer is used to prevent the NAT device from keeping an unused port open.
6. **registration-caching**—Enable for use with all UAs, not just those that are behind NATs. By default, this field is set to **disabled**. If enabled, the Net-Net SBC caches the Contact header in the UA's REGISTER request when it is addressed to one of the following:
 - Net-Net SBC
 - registrar domain value
 - registrar host value

The Net-Net SBC then generates a Contact header with the Net-Net SBC's address as the host part of the URI and sends the REGISTER to the destination defined by the registrar host value.

Whether or not SIP HNT functionality is enabled affects the value of the user part of the URI sent in the Contact header:

- **enabled**—The Net-Net SBC takes the user part of the URI in the From header of the request and appends a cookie to make the user unique. A cookie is information that the server stores on the client side of a client-server communication so that the information can be used in the future.
- **disabled**—The user part of the Contact header is taken from the URI in the From header and no cookie is appended. This is the default behavior of the Net-Net SBC.

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 SBC, which is the Contact address. Then, the Net-Net SBC forwards the request to the Contact-URI it cached from the original REGISTER message.

7. **min-reg-expire**—Set the time in seconds for the SIP interface. The value you enter here sets the minimum registration expiration time in seconds for HNT registration caching. The default value is **300**. The valid range is:

- Minimum—1
- Maximum—999999999

This value defines the minimum expiration value the Net-Net SBC places in each REGISTER message it sends to the real registrar. In HNT, the Net-Net SBC caches the registration after receiving a response from the real registrar and sets the expiration time to the NAT interval value.

Some UAs might change the registration expiration value they use in subsequent requests to the value specified in this field. This change causes the Net-Net SBC to send frequent registrations on to the real registrar.

8. **registration-interval**—Set the Net-Net SBC's cached registration entry interval for a non-HNT endpoint. Enter the expiration time in seconds that you want the Net-Net SBC to use in the REGISTER response message sent back to the UA. The UA then refreshes its registration by sending another REGISTER message before that time expires. The default value is **3600**. The valid range is:

- Minimum—1
A registration interval of zero causes the Net-Net SBC to pass back the expiration time set by and returned in the registration response from the registrar.
- Maximum—999999999

If the expiration time you set is less than the expiration time set by and returned from the real registrar, the Net-Net SBC responds to the refresh request directly rather than forwarding it to the registrar.

Note: With registration caching, there is no NAT; therefore, a short registration interval causes the UA to send excess REGISTER messages.

Although the registration interval applies to non-HNT registration cache entries, and the loosely related NAT interval applies to HNT registration cache entries, you can use the two in combination. Using a combination of the two means you can implement HNT and non-HNT architectures on the same Net-Net SBC. You can then define a longer interval time in the registration interval

field to reduce the network traffic and load caused by excess REGISTER messages because there is no NAT binding to maintain.

9. **route-to-registrar**—Enable routing to the registrar to send all requests that match a cached registration to the destination defined for the registrar host; used when the Request-URI matches the registrar host value or the registrar domain value, not the Net-Net SBC's address. Because the registrar host is the real registrar, it should send the requests back to the Net-Net SBC with the Net-Net SBC's address in the Request-URI. The default value is **disabled**. The valid values are:
 - enabled | disabled

For example, you should enable routing to the registrar if your network uses a Net-Net SBC and needs requests to go through its service proxy, which is defined in the registrar host field.

Global SIP Configuration

To configure the SIP configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# 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. **registrar-domain**—*Optional*. Define the domain to match against the host part of a URI to determine if a request is addressed to the registrar. If there is a match, the registration caching, NAT traversal, and route to registrar parameter values for the SIP interface are applied to the request. By default, this field remains empty. Available values are:
 - an asterisk (*) to specify the values apply to all requests.
 - any alphanumeric character or any combination of alphanumeric characters. For example, acme1.com.

A hostname consists of any number of domain labels, separated by dots (.), and one top label. A top label is the last segment of the hostname. It must start with an alphabetical character. After the first character, a top label can consist of any number or combination of alphanumeric characters, including those separated by dashes. The dash must be preceded and followed by alphanumeric characters. A single alphabetical character is the minimum requirement for a hostname field (for example, c to indicate .com).

When the REGISTER message's Request-URI has an FQDN, it is matched against the registrar domain's value to determine if the message needs to be forwarded to the registrar port on the registrar host. The registrar domain's value is also used when route to registrar is set to enabled, to determine if a request needs to be forwarded to the registrar.

Only the right-hand part of the domain name in the Request-URI needs to match the registrar domain value. For example, acme3.acmepacket.com matches acmepacket.com. However, the entire domain label within the domain name

must match. For example, the domain label “acme3.acmepacket.com” would not match “packet.com”.

5. **registrar-host**—Define the address of the registrar for which requests for registration caching, NAT traversal, and router to registrar options apply. You can use a specific hostname, a IP address, or a wildcard (*):
 - an asterisk (*) indicates normal routing (local policy, DNS resolution, and so on) is used to determine the registrar’s address.
 - hostname: can consist of any alphanumeric character or any combination of alphanumeric characters (for example, acme1. com). The hostname can consist of any number of domain labels, separated by dots (.), and one top label. You can use the minimum field value of a single alphabetical character to indicate the top label value (for example, c to indicate . com).
 - IPv4 address: must follow the dotted notation format. Each of the four segments can contain a numerical value between zero (0) and 255. For example, 192. 168. 201. 2. An example of a invalid segment value is 256. 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. The default value is 0. The valid range is:

- Minimum—0, 1025
- Maximum—65535

The following example shows the values for a single domain and registrar configuration.

```
si p-config
  state          enabled
  operation-mode dialog
  dialog-transparency  disabled
  home-real-m-id    acme
  egress-real-m-id
  nat-mode          public
  registrar-domain
  registrar-host
  registrar-port    0
  init-timer        500
  max-timer         4000
  trans-expire     32
  invite-expire    180
  inactive-dynamic-conn 32
  red-sip-port     1988
  red-max-trans   10000
  red-sync-start-time 5000
  red-sync-comp-time 1000
  last-modified-date 2005-03-19 12:41:28
```

SIP Registration Local Expiration

When you deploy multiple Net-Net SBCs in series and they have registration caching and HNT configured, registration cache entries might expire prematurely in instances with several devices provisioned with the same address of record (AoR). Now you can configure a SIP interface option to prevent the premature expiration.

How It Works

When you use registration caching and HNT, the Net-Net SBC adjusts the expiration time it sends to user agents (UAs) in REGISTER responses based on the registration interval you configure. It can be the case that a SIP user has multiple registered contact endpoints at the UA to which a response is sent. If the URI in the Contact contains the UA's address and that UA included the Contact in the REGISTER request, then the Contact is seen as exclusively belonging to that UA. In the REGISTER response, this Contact (exclusive to the UA) includes the local expiration time, a time based on the SIP interface configuration's registration or NAT interval value. Additional Contacts (not exclusive to the UA) in the REGISTER response have the expiration time from the REGISTER response the registrar sent to the Net-Net SBC.

It is this default behavior can cause registration cache entries to expire prematurely in the Net-Net SBC nearest a registrar when multiple Net-Net SBCs are deployed in series. Multiple registering UAs for a single SIP user, for example, might trigger the early expiration. The SIP you can configure an option per SIP interface that causes the Net-Net SBC to send the local registration expiration time in all in the Expires parameter of all Contact headers included in REGISTER responses sent from the SIP interface.

ACLI Instructions and Examples

You can configure this feature either for the global SIP configuration, or for an individual SIP interface.

To configure SIP registration local expiration for the global SIP configuration:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-config** and press <Enter>. If you are editing an existing configuration, select the configuration so you can enable this feature.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **reg-local-expires** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(sip-config)# options +reg-local-expires
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration's options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

To configure SIP registration local expiration for an individual SIP interface:

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

```
ACMEPACKET# config 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-local-expires** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(sip-interface)# options +reg-local-expires
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

SIP HNT Forced Unregistration

If you use HNT and experience the issue explained in this section, consider using the Net-Net SBC’s forced unregistration feature. When this feature is enabled and a registration entry for an endpoint expires, the Net-Net SBC notifies 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 SBC and the softswitch by deleting a contact binding that is no longer valid or needed
- Preventing extra bindings from being generated at the softswitch (e.g., in instances when the UA or NAT restart)

This feature applies to:

- HNT endpoints with registration caching enabled by default, and when the **nat-traversal** parameter in the SIP interface configuration is set to **always**
- non-HNT endpoints with registration caching enabled, when the registration-interval parameter in the SIP interface configuration is used in the expires header sent to the UA in the 200 OK

When to Use Forced Unregistration

For typical HNT use, it is common that the registration interval between the client UA and the Net-Net SBC is between sixty (60) and one hundred and twenty (120) seconds. This differs significantly from the re-registration interval between the Net-

Net SBC 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 of 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 SBC encodes a cookie in the userinfo of the Contact header in the REGISTER message. This is a function of the source IPv4 address and port from which the request came, i.e., the ephemeral port in the NAT for DSL scenarios. Therefore, additional bindings that remain for long periods of time are created at the registrar if, for example, the:

- UA reboots
- Ethernet link between the UA and the DSL router is lost for over two minutes
- DSL crashes
- DSL/ATM layer between the DSL router

Caution for Using Forced Unregistration

You should use caution when applying SIP HNT forced unregistration for the following reasons:

- It can have an impact on the performance of your Net-Net SBC 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.

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(config)# **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 IP Address Hiding and NATing in XML

Adding to its topology hiding and NAT capabilities, the Net-Net SBC now performs those functions for pertinent IP addresses that are not part of the standard SIP message header format. Previously, such addresses were visible to the next hop in the SIP session path.

Note that this feature adds to the Net-Net SBC's pre-existing ability to perform this function for XML messages; this new support is specifically for the `keyset-info` message type.

How It Works

For incoming SIP NOTIFY messages, the Net-Net SBC searches for the `application/keyset-info+xml` content type in the message. When it finds this content type, it searches further to detect the presence of `<di : remote-uri >` or `<di : local-uri >` XML tags and then NATs the IP addresses in the tags it finds. Specifically, the Net-Net SBC changes:

- The `<di : remote-uri >` IP address to be the egress SIP interface's IP address
- The `<di : local-uri >` IP address to be the IP address of the next hop to which the message is being sent

Sample SIP NOTIFY with NATed XML

The following is a sample SIP NOTIFY message as it might arrive at the Net-Net SBC. Note that it contains the `<di : remote-uri >` or `<di : local-uri >` XML tags on which the system will perform NAT; these lines appear in bold text.

```

NOTIFY sip:15615281021@10.152.128.253:5137;transport=udp SIP/2.0
To: 15615281021
<sip:15615281021@10.152.128.102:5080>;tag=5c93d019904036a
From: <sip:15615281021@10.152.128.102:5080>;tag=test_tag_0008347766
Call-ID: 3215a76a979d0c6
CSeq: 18 NOTIFY
Contact: <sip:15615281021@10.152.128.102:5080;maddr=10.152.128.102>
Via: SIP/2.0/UDP
10.152.128.102:5060;branch=z9hG4bK_brancha_0023415201
Event: keyset-info
Subscription-state: active;Expires=2778
Accept: application/keyset-info+xml
Content-Type: application/keyset-info+xml
Content-Length: 599
Max-Forwards: 70

<?xml version="1.0"?>
<keyset-info xml ns="urn:ietf:params:xml:ns:keyset-info"
version="16"
entity="15615281021">
<ki-data>
<ki-state>active</ki-state>
<ki-event>unknown</ki-event>
</ki-data>
<di:dialog_id="dialog_id_201" call-id="1395216611-1987932283256611-11-0884970552" local-tag="test_tag_0008347790" direction="recipient">
<di:state>trying</di:state>
<di:duration>2778</di:duration>
<di:local-uri>sip:15615281021@10.152.128.253:5137</di:local-uri>

```

```
<di : remote-uri >sip: 1004@10.152.128.102</di : remote-uri >
</di : di al og>
</keyset-i nfo>
```

Once the Net-Net SBC has completed the NAT process, the `<di : remote-uri >` and `<di : local-uri >` XML tags look like this

```
<di : local-uri >sip: 15615281021@192.168.200.99: 5137</di : local-uri >
<di : remote-uri >sip: 1004@192.168.200.49</di : remote-uri >
```

because egress the SIP interface's IP address is 192.168.200.49 and the next hop's IP address is 192.168.200.99.

ACLI Instructions and Examples

This feature does not require any configuration.

SIP Server Redundancy

This section explains how to configure SIP server redundancy. SIP server redundancy involves detecting that an upstream/downstream SIP signaling entity has failed, and adapting route policies dynamically to remove it as a potential destination.

Overview

You establish SIP server redundancy by creating session agents, which are virtual representations of the SIP signaling entities. These agents are then collected into a session agent group, which is a logical collection of two or more session agents that behaves as a single aggregate entity. 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 SBC's IPv4 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 SBC'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 SBC. To accomplish this, an option has been added to the local policy that causes the next_hop value to be used as the Request-URI and prevents the addition of Route headers. This option is the replace_uri value in the local policy.

The following table lists the policy attributes for the local policy:

Parameter	Description
next hop	IP address of your internal SIP proxy. This value corresponds to the IP address of the network interface associated with the SIP proxy.
realm	Number of the port associated with the SIP port.
replace uri	Stores the transport protocol used for sending and receiving signaling messages associated with the SIP port.
allow anonymous	Indicates whether this SIP port allows anonymous connections from session agents.

Note: You should also define the ping method intervals for the session agents so that the Net-Net SBC can detect when the agents are back in service after failure.

For more information about local policy, see the [Session Routing and Load Balancing \(743\)](#) 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)#[/b]

3. Type **policy-attributes** and press <Enter>. The system prompt changes.

```
ACMEPACKET(I ocal -pol i cy)# pol i cy-attribu tes
ACMEPACKET(I ocal -pol i cy-attribu tes)#[/pre]

```

From this point, you can configure policy attributes for the local policy. To see all local policy attribute options, enter a ? at the system prompt.

4. **action**—Set this parameter to **replace-uri**, which causes NAT rules such as realm prefixing to be overridden. The default value is **none**. Valid values are:

- none | replace-uri | redirect

The replace URI field allows you to indicate whether the local policy's value is used to replace the Request-URI in outgoing requests. This boolean field can be set to either enabled or disabled.

Administratively Disabling a SIP Registrar

The Net-Net SBC's registration cache feature is commonly used to support authorization. It also allows the Net-Net SBC to respond directly to SIP REGISTER requests from endpoints rather than forwarding every REGISTER message to the Registrar(s). In the Net-Net SBC, Registrars are frequently configured as session agents, and an association between each endpoint and its Registrar is stored with the registration cache information.

In Release 4.0.1 and later, the **invalidate-registrations** parameter in the session agent configuration enables the Net-Net SBC to detect failed Registrar session agents and automatically forward subsequent REGISTER requests from endpoints to a new Registrar. You can now perform the same behavior manually through a new ACLI command. When you use this command, the Net-Net SBC acts as though the registrations have expired.

How It Works

For each SIP session agent, you can enable the manual trigger command, and then use the command from the main Superuser ACLI prompt. The **reset session-agent** command provides a way for you to send a session agent offline. Session agents can come back online once they send 200 OK messages the Net-Net SBC receives successfully.

Without using the manual trigger, session agents can go offline because of they do not respond to pings or because of excessive transaction timeouts. However, you might not want to use these more dynamic methods of taking session agents out of service (and subsequently invalidating any associated registrations). You can disable both of these mechanisms by setting the following parameters to 0:

- **ping-interval**—Frequency (amount of time in seconds) with which the Net-Net SBC pings the entity the session agent represents)
- **ttr-no-response**—Amount of time to wait changing the status of a session agent after it has been taken out of service because of excessive transaction timeouts

However, you can still use the new SIP manual trigger even with these dynamic methods enabled; the trigger simply overrides the configuration to send the session agent offline.

Considerations for Implicit Service Route Use

When implicit service route support is enabled for a SIP interface (in IMS applications), the Net-Net SBC stores the Service Route URIs from the Service-Route headers that are included in 200 OK responses to REGISTER messages.

Subsequently, and even when a session agent is rendered invalid, re-REGISTER messages follow the route stored in the cache instead of using the one defined in the Net-Net SBC.

However, you might not want to use this behavior when you send session agents offline. If you instead want use the route defined in the Net-Net SBC, then you need to configure the SIP interface option called **route-register-no-service-route**.

ACLI Instructions and Examples

This section shows you how to enable the manual trigger for sending session agents out of service, and how to then use the trigger from the command line. This section also shows you how to verify that you have successfully put a session agent out of service.

To enable a SIP session agent to manually trigger it to go out of service:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

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

```

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

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

```

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

4. **invalidate-registrations**—Set this parameter to enabled if you want to use the manual trigger to send this session agent offline (and therefore invalidate the registrations associated with it). The default is disabled.
5. Save and activate your configuration.

To use the manual trigger that sends session agents offline:

1. Note the hostname value (typically the IP address of the endpoint) for the session agent you want to put out of service. You use this name as an argument in the ACLI command to use the manual trigger.

2. At the Superuser prompt, type **reset session-agent**, a <Space>, and the hostname value for the session agent. Then press <Enter>.

```
ACMEPACKET# reset session-agent 192.168.20.45
```

If you enter a session agent that does not exist, the system notifies you that it cannot carry out the reset.

To confirm that a session agent has been sent offline:

You can use the **show sipd endpoint-ip** command to see information confirming that the session agent you sent offline is in that state. The display for this command shows the session agent name or IP address when—and only when—the session agent is enabled to respond to the manual trigger.

After a session agent has been configured to support the manual trigger, you will see either:

- The asterisk (*)—Showing that the session agent is enabled for invalidation, but is still online
- The capital letter X—Showing that the session agent has been sent offline using the manual trigger

The following example uses the session agent IP address 172.16.0.66. In this first sample, the session agent has the **invalidate-registrations** parameter enabled but is still in service:

```
ACMEPACKET# show s1 pd endpoint-ip 9
User <si p: 9580000001@192. 168. 201. 86>
Contact ID=1008 exp=597
UA-Contact: <si p: si pp@192. 168. 200. 254: 9004> UDP
real m=si p192 l ocal =192. 168. 201. 86: 5060 UA=192. 168. 200. 254: 9004 SA=172. 16. 0. 66 *
SD-Contact: <si p: 9580000001-hchse0j m171u2@172. 16. 10. 86: 5060> real m=si p172
Call -ID: 1-20622@192. 168. 200. 254'
Service-Route='<si p: 172. 16. 0. 66: 5060; lr>'
```

Note that the asterisk (*) appears next to the SA value.

The next sample shows the same command carried out, but this time the manual trigger has been used—as shown by the appearance of the X next to the SA value.

```
ACMEPACKET# show s1 pd endpoint-ip 9
User <si p: 9580000001@192. 168. 201. 86>
Contact ID=1008 exp=597
UA-Contact: <si p: si pp@192. 168. 200. 254: 9004> UDP
real m=si p192 l ocal =192. 168. 201. 86: 5060 UA=192. 168. 200. 254: 9004 SA=172. 16. 0. 66 X
SD-Contact: <si p: 9580000001-hchse0j m171u2@172. 16. 10. 86: 5060> real m=si p172
Call -ID: 1-20622@192. 168. 200. 254'
Service-Route='<si p: 172. 16. 0. 66: 5060; lr>'
```

SIP Distributed Media Release

This section explains how to configure distributed media release (DMR). SIP DMR lets you choose whether to include multi-system (multiple Net-Net SBCs) 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 SBC; even if the SIP signaling messages traverse multiple Net-Net SBCs. It encodes IPv4 address and port information for the media streams described by the media, for example SDP.

With SIP DMR, the media realm and IPv4 address and port information from the UA's SDP is encoded into SIP messages (either in the SIP header or in the SDP) as they enter the backbone network. The information is decoded by a Net-Net SBC 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 SBCs 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 SBCs.

You can enable this feature on a per-realm basis and multiple realms can be supported.

Endpoint Locations

You can configure the Net-Net SBC 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 SBC is only part of the media path for traffic that originates and terminates in different networks.

If configured to do so, the Net-Net SBC can release media:

- Between endpoints supported by a single Net-Net SBC
 - In the same network/VPN
 - In the same network behind the same NAT/firewall
- Between endpoints supported by multiple distributed Net-Net SBCs
 - 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, `medi a-rel ease=sdp; acme-medi a`). The encoded data is placed into an `acme-medi a` attribute in the SDP:

`a=acme-medi a: <encoded-medi a-interface-info>`

- SIP header parameter

Media release data can be placed in a header parameter of a SIP header (for example, `medi a-rel ease=Contact; acme-medi a`). The encoded data is placed into an `acme-medi a` parameter in the Contact header:

`Contact: <si p: 1234@abc.com>; acme-medi a=<encoded-medi a-interface-info>`

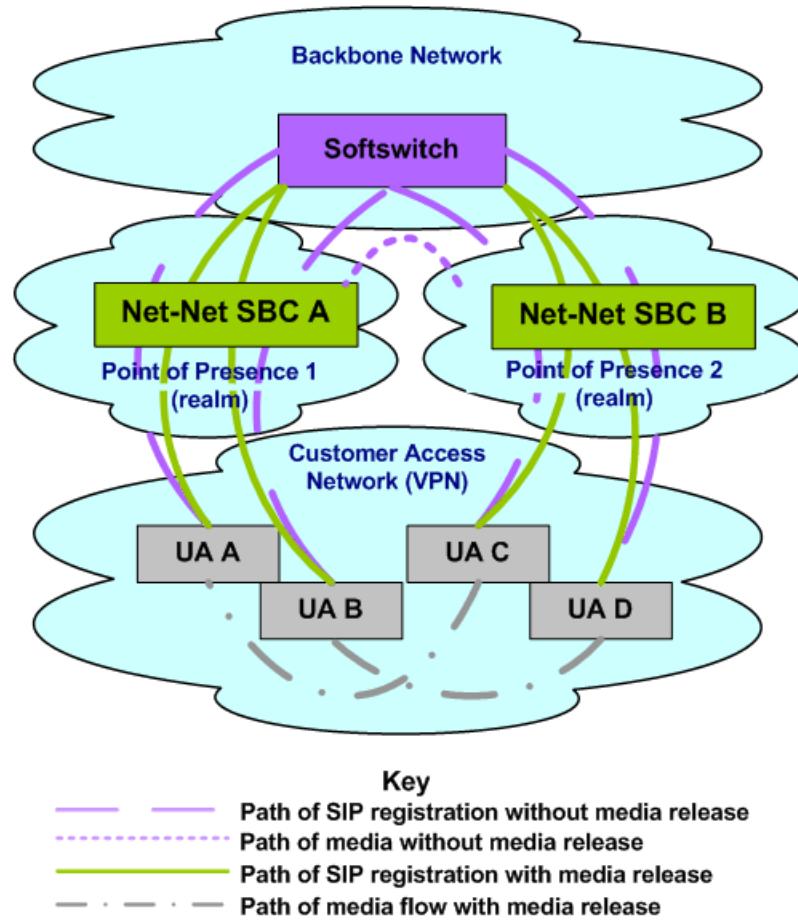
- SIP header

Media release data can appear in a SIP header (for example, `medi a-rel ease=P-Medi a-Rel ease`). The encoded data is placed into a `P-Medi a-Rel ease` header:

`P-Medi a-Rel ease: <encoded-medi a-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 SBCs.



Overview of SIP DMR Configuration

To configure SIP DMR:

1. Edit the SIP config element's option field.

The `media-release="<header-name>[;<header-param>]"` option defines how the SIP distributed media release feature encodes IPv4 address and port information. If the `media-release` parameter is configured in the options field but no header is specified, the parameter value of `P=Media-Release` will be used. This parameter is optional and is not configured by default.

2. Enable SIP DMR for the entire realm by setting the realm config element's msm release field to enabled.

The media IPv4 address and port information is encoded into outgoing SIP messages and decoded from incoming SIP messages for all of the realms (in each realm-config element) with which the SIP distributed media release will be used.

Note: You can also use the realm config element's mm in network field to release the media back to a connected network that has multiple realms. This field is not specific SIP distributed media release and it is not required for the SIP DMR to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Net-Net SBC release the media.

ACLI Instructions and Examples

To configure media release:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# **session-router**
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-config**
ACMEPACKET(sip-config)#
From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.
4. Type **options** followed by a <Space>.
5. After the <Space>, type the media release information in the following format:
media-rel ease=" <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-rel ease=P-Media-Rel ease
```

6. Press <Enter>.

Note: If the media-release parameter is configured in the options field, but no header is specified, then the parameter value of P-Media-Rel ease will be used. P-Media-Rel ease 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.

```
mediarel ease="P-Media-Rel ease"  
mediarel ease="Contact; acme-media"  
mediarel ease="sdp; acme-media"
```

Configuring the Realm Configuration

You need to set the each realm config element's `msm` release field to enabled for all the realms for which you want to use SIP DMR.

Although the `mm` in network field is not specific to the SIP distributed media release feature, it can be used to release the media back to a connected network that has multiple realms. This field does not need to be configured in order for the SIP distributed media release feature to work. However, if this field is set to enabled and the ingress and egress realms are part of the same network interface, it lets the Net-Net SBC release the media.

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

```
ACMEPACKET# configure terminal
```

2. Type `media-manager` and press <Enter> to access the media-related configurations.

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

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

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

```
ACMEPACKET(realm-config)#

```

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

4. **msm-release**—Enable DMR within this realm on this Net-Net SBC. The default value is `disabled`. The valid values are:
 - enabled | disabled
5. Repeat for each realm on which you want to enable DMR.

Add-On Conferencing

This section explains how to configure the add-on conferencing functionality. It also includes a description of the SIP B2BUA functionality related to the SIP add-on conferencing. This description includes information about Contact header mapping and processing and Refer-to header processing.

Overview

SIP add-on conferencing lets you:

- Use the Net-Net SBC's add-on conferencing feature for network architectures in which the conference initiator is located on a different network than that of the media server.
- Configure the Net-Net SBC to enable Contact header mapping for the Refer-To header.

Caveats

The following caveats are associated with add-on conferencing:

- Contact header mapping is not replicated on the standby Net-Net SBC in an HA Net-Net SBC pair architecture.
- Upon switchover, any conferences in progress remain in progress, but no new parties can be invited to or join the conference.
- By default, the Net-Net SBC does not map SIP Contact headers for reasons of performance.

Add-On Conferencing Scenario

The add-on conferencing scenario described in the following example applies to a network architecture involving the Net-Net SBC and a media server that is located on a different network from the other conference participants. In this scenario, the Net-Net SBC resides on a standalone network that connects two additional, separate networks.

Some network architectures have a media server on a different network from the one on which the phones reside. In this scenario, all requests and/or responses going from the phones (Phone A, Phone B, or Phone C) to Media Server D and vice versa are translated according to their corresponding SIP-NAT. All headers subjected to NAT are encoded and decoded properly as they traverse the Net-Net SBC, except for the Contact header. This exception occurs because the SIP process on the Net-Net SBC runs as a SIP B2BUA and not as a SIP proxy.

The SIP B2BUA re-originate the Contact headers of the User Agents (UAs) participating in SIP sessions with local Contact headers to make sure that they receive all future in-dialog requests. For an in-dialog request, the B2BUA can identify the dialog and find the Contact URI of the other leg of the call.

The Net-Net SBC add-on conferencing feature applies to situations when the Contact URI is used in another dialog. In such a case, the SIP B2BUA will not be able to find the correct dialog that retrieves the correct Contact URI of the other leg if it needs to replace the Contact URI.

Using the SIP add-on conferencing, the SIP B2BUA on the Net-Net SBC can map the Contact headers it receives to the Contact headers it creates. It can also convert the Refer-To URI to the correct value required for forwarding the REFER request.

SIP B2BUA Functionality

This section describes the role of the Net-Net SBC's SIP B2BUA in the add-on conferencing scenario that requires Contact header mapping for the Refer-To header.

When the Net-Net SBC starts up, the SIP B2BUA reads and parses the list of options in the SIP configuration. If the refer to uri prefix is an appropriate value (it is not an empty string), the Net-Net SBC will have a text prefix value the media server can use to denote a conference ID in its Contact header. With this information, the SIP B2BUA sets up a Contact header mapping.

You configure the Net-Net SBC to enable Contact header mapping for the Refer-To header by editing the SIP config options parameter. The SIP B2BUA on the Net-Net SBC can then map the Contact headers it receives to the Contact headers it creates.

Contact Header Processing

The Contact header mapping matches a Contact header that contains the refer to URI prefix to the corresponding Contact header that the Net-Net SBC's SIP B2BUA re-originate. Contact headers that do not contain the refer to URI prefix are not mapped (so that performance of the Net-Net SBC is minimally affected).

Only the Contact header in an INVITE request and its 200 OK response are checked for the refer to URI prefix and added to the Contact header mapping. Contact headers appearing in other SIP requests/responses are not checked.

Target Mapping and Conferences

If the Net-Net SBC is configured to enable Contact header mapping for the Refer-To header, then Contact header target maps are established for each individual call. The Net-Net SBC's SIP B2BUA uses these maps to allow the media server to connect the conference initiator with the conference-in parties.

Prior to terminating the call (hanging up), the conference initiator can contact other parties and invite those additional parties to join the conference. These other parties can join the existing conference because the target mapping for the conference is still in effect on the Net-Net SBC.

Once the conference initiator hangs up, the Net-Net SBC discards the mapping from the conference.

Refer-To Header Processing

When a Refer-To header is present in a REFER request that arrives at the SIP B2BUA after the incoming request is properly translated according to its SIP-NAT, the SIP B2BUA follows these steps:

1. The SIP B2BUA parses the Refer-To URI.
2. If the user part of the Refer-To URI contains the refer to URI prefix, the SIP B2BUA searches the Contact header mapping for a match of the user part of the URI.
If the user part of the Refer-To URI does not contain the refer to URI prefix, the SIP B2BUA leaves the existing Refer-To URI unchanged.
3. If the user part of the Refer-To URI contains the refer to URI prefix and a match of the Refer-To URI is found, the SIP B2BUA replaces the existing Refer-To URI with the URI of the corresponding Contact URI stored in the matched record. This replacement enables the NAT function to properly decode the replacement URI and change it back to the form originally received by the Net-Net SBC. As a result, the correct conference ID is restored in the Refer-To header prior to the request being sent to its next hop.
If the user part of the Refer-To URI contains the refer to URI prefix but a matched URI cannot be found, the SIP B2BUA will leave the existing Refer-To URI unchanged and will write a WARNING level log message to record the failure.

ACLI Instructions and Examples

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config ure terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# sessi on-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)# si p-confi g
ACMEPACKET(si p-confi g) #
From this point, you can configure SIP config parameters. To view all SIP config parameters, enter a ? at the system prompt.
4. Type **options** followed by a <Space>.
5. After the <Space>, type the add-on conferencing information in the following format:
options refer-to-uri -prefi x="conf"
For example:
ACMEPACKET(si p-confi g) # options refer-to-uri -prefi x="conf"
6. Press <Enter>.

SIP REFER Method Call Transfer

In prior releases, the Net-Net SBC supports the SIP REFER method by proxying it to the other UA in the dialog. A handling mode has been developed for the REFER method so that the Net-Net SBC automatically converts a received REFER method into an INVITE method, thus allowing the Net-Net SBC to transfer a call without having to proxy the REFER back to the other UA.

This function can be configured for a specified SIP interface, a realm, or a session agent. When all three elements have the SIP REFER method call transfer functionality configured, the session-agent configuration takes precedence over realm-config and sip-interface configurations. If session-agent is not configured, and realm-config and sip-interface are, realm-config takes precedence.

How it Works

The Net-Net SBC has a configuration parameter giving it the ability to provision the handling of REFER methods as call transfers. The parameter is called `refer-call-transfer`. When this feature is enabled, the Net-Net SBC creates an INVITE message whenever it receives a REFER. The Net-Net SBC sends this INVITE message to the address in the Refer-To header. Included in the INVITE message is all the unmodified information contained in the REFER message. The previously negotiated codec is also still used in the new INVITE message. NOTIFY and BYE messages are sent to the UA upon call transfer completion.

If a REFER method is received containing no Referred-By header, the Net-Net SBC adds one, allowing the Net-Net SBC to support all call agent screen applications.

In addition, the SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers
- Both successful and unsuccessful call transfers
- Early media from the Referred-To party to the transforee
- REFER method transfer from different sources within the destination realm
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.
- The Referred-To party can be both in a different realm (and thus a different steering pool) from the referrer, and in the same realm
- The associated latching should not prohibit the Referred-To party from being latched to while the referee is still sending media.

Unsuccessful Transfer Scenarios

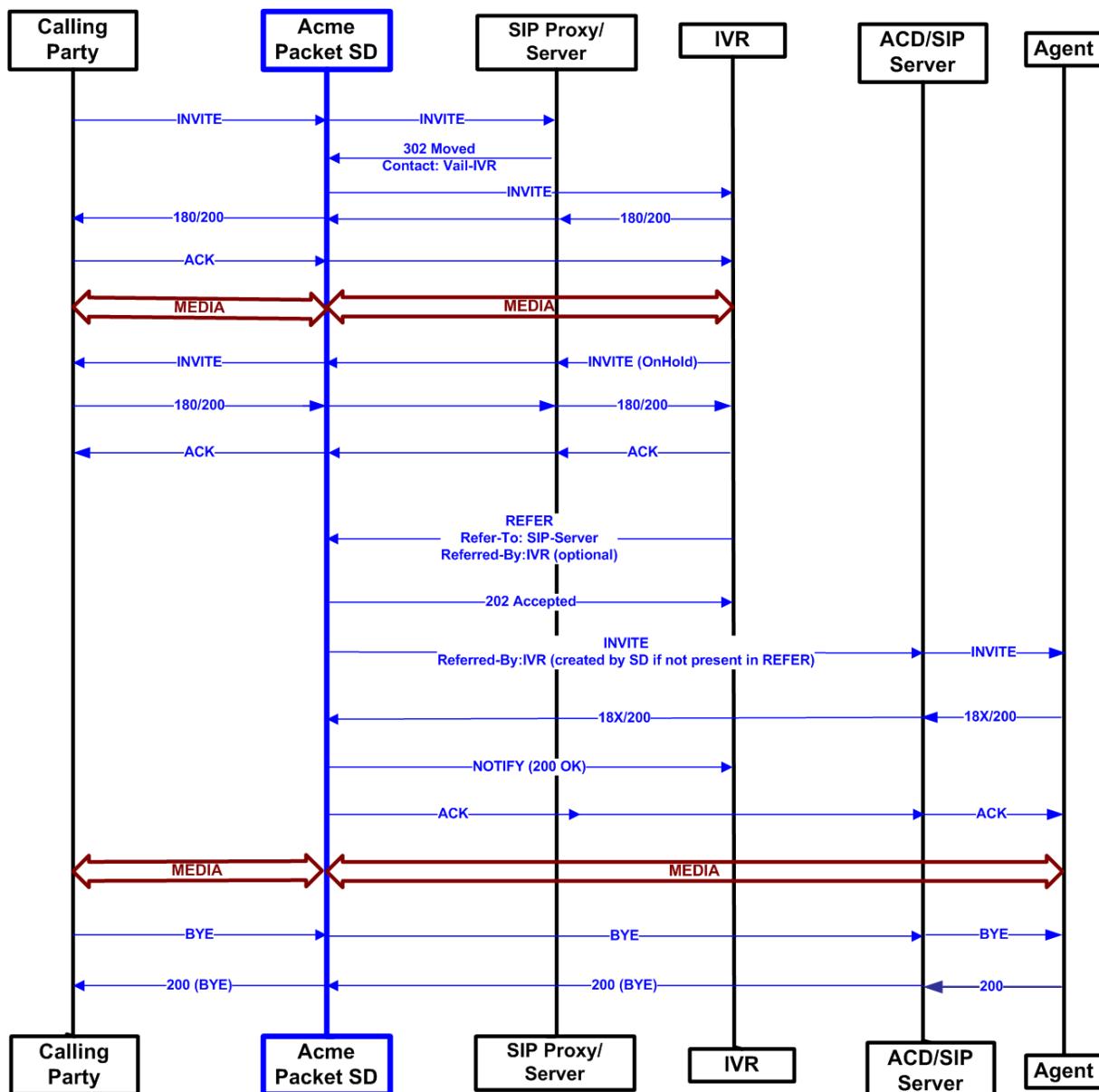
The Net-Net SBC does not successfully handle the following failed, unusual, and unexpected transfer scenarios:

- The new INVITE to the Referred-To party gets challenged, the Net-Net SBC does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Net-Net SBC.

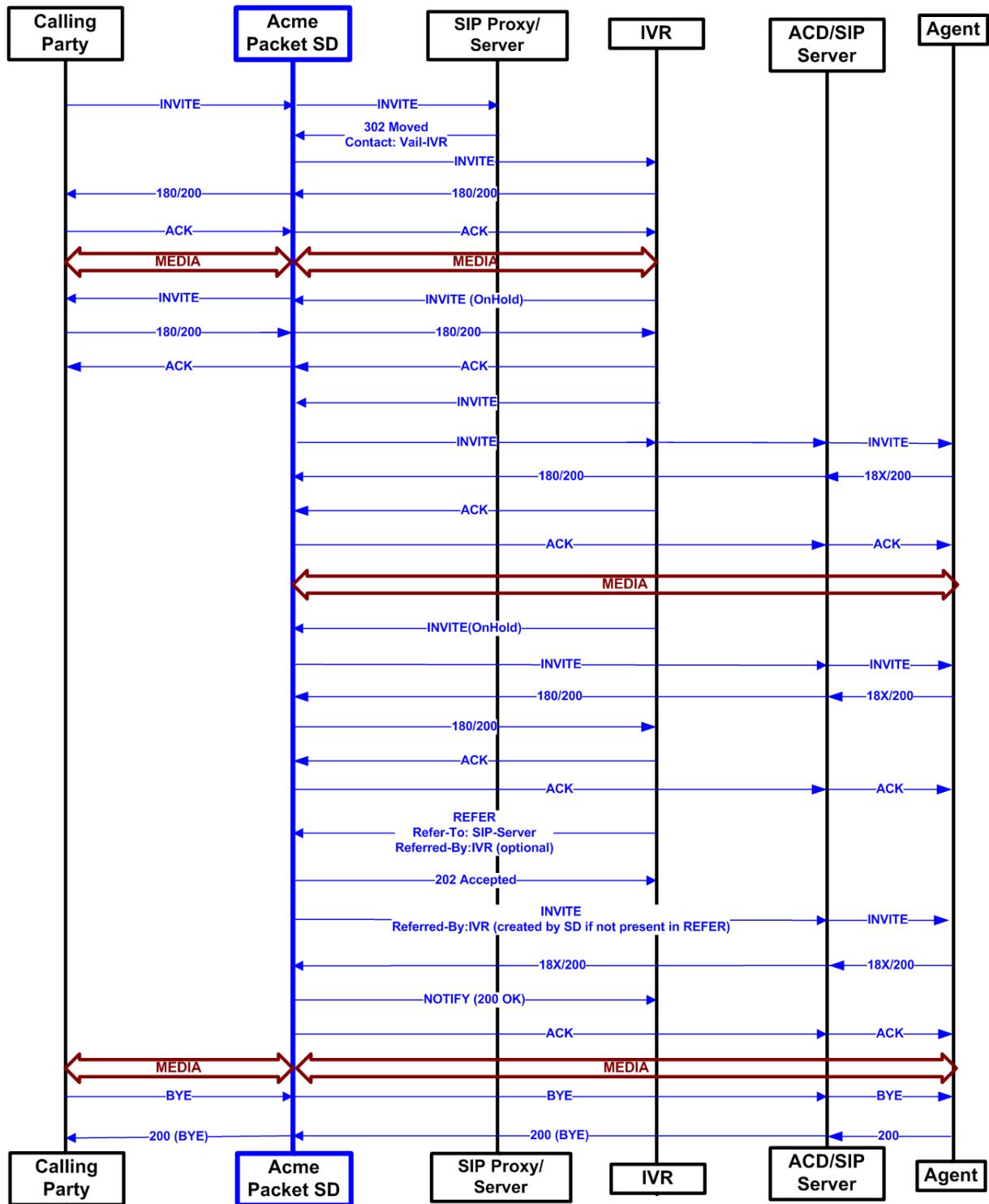
- The Net-Net SBC shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Net-Net SBC ignores any REFER method containing a MIME attachment.
- The Net-Net SBC recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.
- The original parties agreed on a codec using a dynamic payload type, and the Referred-To party happens to use a different dynamic payload number for that codec.

Call Flows

The following is an example call flow for an unattended call transfer:



The following is an example call flow of an attended call transfer:



ACLI Instructions and Examples

To enable SIP REFER method call transfer in the realm-config:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **config 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(real m-config)#
4. **refer-call-transfer**—Set to **enabled** to enable the refer call transfer feature. The default for this parameter is **disabled**.
5. Save and activate your configuration.

To enable SIP REFER method call transfer in the sip-interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **config 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. **refer-call-transfer**—Set to **enabled** to enable the refer call transfer feature. The default for this parameter is **disabled**.
5. Save and activate your configuration.

To enable SIP REFER method call transfer in a realm:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **config 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(real m-config)#
4. **refer-call-transfer**—Set to **enabled** to enable the refer call transfer feature. The default for this parameter is **disabled**.
5. Save and activate your configuration.

To enable SIP REFER method call transfer in the session-agent:

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

- ```

ACMEPACKET# config 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(media-manager)# session-agent
ACMEPACKET(session-agent)#
4. refer-call-transfer—Set to enabled to enable the refer call transfer feature. The default for this parameter is disabled.
5. Save and activate your configuration.

```

## REFER-Initiated Call Transfer

In prior releases, the Net-Net SBC supported REFER-initiated call transfer either by proxying the REFER to the other User Agent in the dialog, or by terminating the received REFER and issuing a new INVITE to the referred party. These static alternate operational modes could be configured for specific SIP interfaces, realms, or session agents.

Release S-C6.2.0 enhances support with an additional operational mode that determines on a call-by-call basis whether to proxy the REFER to the next hop, or terminate the REFER and issue an INVITE in its stead.

**Note:** With the release of Version S-C6.2.0, support for REFER-initiated call transfer is no longer available for SIP interfaces; support must be configured for realms and/or session agents.

## How it Works

Version S-C6.2.0 provides a new configuration parameter **dyn-refer-term**, and a revised **refer-call-transfer** parameter (both available in **realm-config** configuration mode) that specify call transfer modes.

With the **refer-call-transfer** parameter set to **disabled** (the default), all received REFERs are simply proxied to the peer User Agent.

With the **refer-call-transfer** parameter set to **enabled**, the Net-Net SBC terminates all REFERs, generates a new INVITE, and sends the INVITE to the address in the Refer-To header.

With the **refer-call-transfer** parameter set to **dynaminc** (a new value introduced with Version S-C6.2.0), the Net-Net SBC determines REFER handling on a call-by-call basis as follows:

1. Check the **refer-call-transfer** value for the session agent from which the REFER was received, or for ingress realm (the realm that received the REFER).
  - If the value is **disabled**, proxy the REFER to the peer User Agent, to complete REFER processing.
  - If the value is **enabled**, terminate the REFER and issue an new INVITE to the referred party, to complete REFER processing.
  - If the value is **dynaminc**, identify the next hop session agent or the egress realm.

2. Check the **dyn-refer-term** value for the next hop session agent, or for the egress realm.
 

If the **dyn-refer-term** value is **disabled** (the default), proxy the REFER to the next hop to complete REFER processing.

If the **dyn-refer-term** value is **enabled**, terminate the REFER and issue a new INVITE to the referred party to complete REFER processing

## Supported Scenarios

In the basic scenario for REFER initiated call transfer, a call is established between two User Agents (Alice and Bob). User Agent Bob then sends a REFER request to transfer the call to a third User Agent Eva. With dynamic call-transfer enabled, the Net-Net SBC prevents the REFER from being sent to Alice and generates the INVITE to Eva.

If the INVITE to Eva succeeds, the Net-Net SBC sends a re-INVITE to Alice modifying the SIP session as described in Section 14 of RFC 3261, *SIP: Session Initiation Protocol*. At this point the Net-Net SBC cancels the original dialog between the Net-Net SBC and Bob.

If the INVITE to Eva fails, call disposition depends on whether or not Bob issued a BYE after the REFER call transfer. If the Net-Net SBC did receive a BYE from Bob (for instance, a blind transfer), it proxies the BYE to A. Otherwise, the Net-Net SBC retains the original SIP session and media session, thus allowing Bob to re-establish the call with Alice by sending a re-INVITE. In this case, the Net-Net SBC sets a timer (32 seconds), after which a BYE will be sent.

If a REFER method is received containing no Referred-By header, the Net-Net SBC adds one, allowing the Net-Net SBC to support all call agent screen applications.

In addition, the SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers
- Both successful and unsuccessful call transfers
- Early media from the Referred-To party to the transforee
- REFER method transfer from different sources within the destination realm
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.
- The Referred-To party can be both in a different realm (and thus a different steering pool) from the referrer, and in the same realm
- The associated latching should not prohibit the Referred-To party from being latched to while the referee is still sending media.

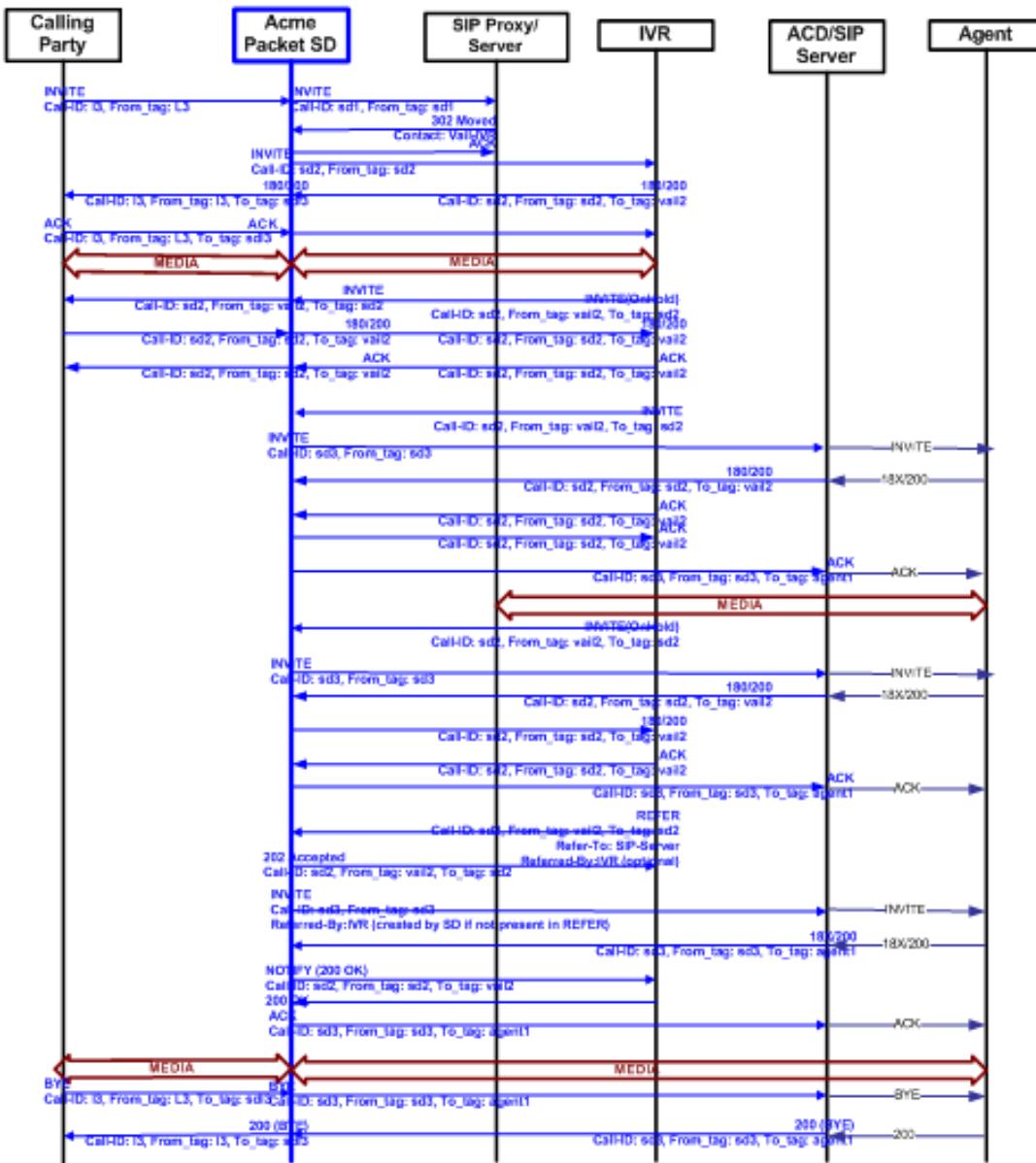
The Net-Net SBC does not successfully handle the following anomalous transfer scenarios:

- The new INVITE to the Referred-To party gets challenged — the Net-Net SBC does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.

- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Net-NET SBC.
- The Net-Net SBC shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Net-Net SBC ignores any REFER method containing a MIME attachment.
- The Net-Net SBC recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.

## Call Flows

The following is an example call flow for an unattended call transfer:



The following is an example call flow of an attended call transfer:



## REFER Source Routing

If, after the conclusion of static or dynamic REFER handling, the REFER is terminated and a new INVITE issued, users now can specify a policy lookup behavior based upon either the source realm of the calling party (the INVITE originator), or the source realm of the referring party (the REFER originator).

Behavior is controlled by a new **refer-src-routing** parameter in the **sip-config** configuration element.

**disabled**, the default value, specifies that the Net-Net SBC performs a policy lookup based on the source realm of the calling party.

**enabled** specifies that the Net-Net SBC performs a policy lookup based on the source realm of the referring party.

## ACLI Instructions and Examples

### To enable realm-based REFER method call transfer:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.  

```
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>.  

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(real-m-config)#
```
4. **refer-call-transfer** — Retain the default (**disabled**) to proxy all REFERs to the next hop. Use **enabled** to terminate all REFERs and issue a new INVITE. Use **dynamic** to specify REFER handling on a call-by-call basis, as determined by the value of the **dyn-refer-term** parameter.
5. **dyn-refer-term** (meaningful only when **refer-call-transfer** is set to **dynamic**) — Retain the default (**disabled**) to terminate the REFER and issue a new INVITE. Use **enabled** to proxy the REFER to the next hop.
6. Save and activate your configuration.

### To enable session-agent-based REFER method call transfer:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **session-agent** and press <Enter>.  

```
ACMEPACKET(media-manager)# session-agent
ACMEPACKET(session-agent)#
```
4. **refer-call-transfer** — Retain the default (**disabled**) to proxy all REFERs to the next hop. Use **enabled** to terminate all REFERs and issue a new INVITE. Use **dynamic** to specify REFER handling on a call-by-call basis, as determined by the value of the **dyn-refer-term** parameter.

5. **dyn-refer-term** (meaningful only when **refer-call-transfer** is set to dynamic) — Retain the default (**disabled**) to terminate the REFER and issue a new INVITE. Use **enabled** to proxy the REFER to the next hop.
6. Save and activate your configuration.

**To specify policy lookup for a newly generated INVITE:**

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(configure)# **sip-config**  
ACMEPACKET(sip-config)#
4. **refer-src-routing** — Retain the default (**disabled**) to perform a policy lookup based upon the source realm of the calling party (the issuer of the original INVITE). Use **enabled** to perform a policy lookup based upon the source realm of the referring party (the issuer of the REFER).  
ACMEPACKET(configure)# **refer-src-routing**  
ACMEPACKET(configure)#
5. Save and activate your configuration.

## SIP REFER: Re-Invite for Call Leg SDP Renegotiation

---

Enhancing the original implementation of SIP REFER termination introduced in Release S-C6.0.0, this change to Net-Net SBC behavior allows for SDP renegotiation between both parties of a transferred call.

### Scenario

In a call transfer initiated by SIP REFER, a call is established between two user agents, UA-A and UA-B. UA-B then sends a REFER request to transfer the call to UA-C. The challenge is that UA-A and UA-B had already been communicating using mutually agreed-on codec, while UA-C might not be using an entirely different codec.

To solve this problem, the Net-Net SBC causes a new SIP session and new media session to be created between UA-A and UA-C. The Net-Net SBC removes any resources allocated for use between UA-A and UA-B, and then severs its connection with UA-B. The session between UA-A and UA-C continues.

## SIP Roaming

---

This section explains how to configure SIP roaming. SIP roaming lets subscribers move from one active SIP device to another (at the same site or multiple sites) and retain service at the last registering device.

### Overview

The Net-Net SBC supports multiple active registrations for the same user. The softswitch makes decisions regarding the current location of the user and the handling of requests from devices that are not currently identified as the user location. When there are multiple NATs, the Net-Net SBC is still required to let the softswitch be able to differentiate it.

The Net-Net SBC's SIP roaming ability supports the following features:

- Multiple active registrations from the same user can be cached, allowing subscribers to move from one active SIP device to another (at the same site or multiple sites) and still retain service at the last registering device. With the SIP roaming feature, one person, using multiple devices, can be contacted at all of the devices. These multiple devices (with their unique contact information) register to indicate that they are available for anyone that wants to contact that one person.
- The Net-Net SBC can also inform network devices (such as softswitches) of private SIP device IPv4 addresses (endpoints) and the public firewall address of the user location.

## Process Overview

Caller 1 wants to contact Person A. Caller 1 sends a message to persona@acmepacket.com, but Person A has configured more than one SIP-enabled device to accept messages sent to that address. These devices have unique addresses of desk@10.0.0.4 and phone2@10.0.0.5. Person A has desk@10.0.0.4 and phone2@10.0.0.5 registered with the Net-Net SBC for anything addressed to persona@acmepacket.com.

With the SIP roaming feature, the Net-Net SBC accepts and stores both registrations for persona@acmepacket.com. That way, when someone wants to get in touch with Person A, the messages are sent to both devices (desk@10.0.0.4 and phone2@10.0.0.5) until Person A answers one of them. You do not need to configure your Net-Net SBC for this functionality; your Net-Net SBC automatically provides it.

## Using Private IPv4 Addresses

In addition to supporting multiple registries, the Net-Net SBC can also distinguish user locations by their private IPv4 address and the IPv4 address of the public firewall. Using this information, the Net-Net SBC adds private endpoint and public firewall information to Contact headers.

For example, entering this information causes a Contact header that formerly appeared as the following:

Contact: <si p: 0274116202@63. 67. 143. 217>

to subsequently appear as the following:

Contact: <si p: 0274116202@63. 67. 143. 217; ep=192. 168. 1. 10; fw=10. 1. 10. 21>

The Net-Net SBC's SIP proxy reads this information and populates the contact-endpoint and contact-firewall fields with the appropriate values.

## Example 1: With a NAT Firewall

The Net-Net SBC SIP proxy is configured with the following changeable parameters:

- endpoint= IP address of the SIP UA
- useradd= IP address of the Firewall Public IP address or the source layer 3 IP address of Register message
- userport= IP address port number of the Firewall Public IP address or the source layer 3 IP address port of Register message
- Net-Net SBC address=63.67.143.217
- firewall public address=10.1.10.21
- firewall public address port=10000
- SIP endpoint behind firewall=192.168.1.10

SIP message Contact header:

```
Contact: <si p: 0274116202@63. 67. 143. 217; endpoint=192. 168. 1. 10;
useradd=10. 1. 10. 21; userport=10000; transport=udp>
```

### **Example 2: Without a NAT Firewall**

The Net-Net SBC SIP proxy is configured with the following changeable parameters:

- useradd= IP address of the SIP UA or the source layer 3 IP address of Register message
- userport= IP address port number of the SIP UA or the source layer 3 IP address port of Register message
- Net-Net SBC address=63.67.143.217
- SIP endpoint=192.168.1.10
- SIP endpoint IP address port=5060

SIP message Contact header:

```
Contact: <si p: 0274116202@63. 67. 143. 217; useradd=192. 168. 1. 10;
userport=5060; transport=udp>
```

For SIP, the softswitch responsibility is that the URI SD put in the Contact of the REGISTER message should be reflected in the 200-OK response to the REGISTER request. The Contact header of the response should have an expires header parameter indicating the lifetime of the registration.

The following example shows a Net-Net SBC Send:

```
Contact: <sep: 0274116202@63. 67. 143. 217 endpoint=192. 168. 1. 10;
useradd=10. 1. 10. 21; userport=10000>;
```

The following examples shows the softswitch Respond:

```
Contact: <sep: 0274116202@63. 67. 143. 217 endpoint=192. 168. 1. 10;
useradd=10. 1. 10. 21; userport=10000>; expires=360
```

The contact field for endpoint and firewall parameters only appear in the following:

- Contact header of a REGISTER request sent from the Net-Net SBC to the softswitch server
- Contact header of a REGISTER response sent from the softswitch server to the Net-Net SBC
- Request-URI of an initial INVITE sent from the UT CSA server to the Net-Net SBC

An active endpoint is deleted when it does not register within the registration-interval setting or receives a 401 Unauthorized.

## **ACLI Instructions and Examples**

You can configure the SIP configuration's options parameter to indicate that you want to use the *private* IP address of the SIP device that the user is using and/or the *public* firewall address that identifies the location of the device. If defined, these options will be added as parameters to all Contact headers.

You can identify the endpoint and/or firewall information using the following options:

- contact-endpoint=<value> where <value> is the endpoint address or label
  - contact-firewall=<value> where <value> is the firewall address or label
1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# config terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

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

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

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

```
ACMEPACKET(sip-config)#

```

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

4. Type **options** followed by a <Space>.
5. After the <Space>, type the information for an endpoint or a firewall, or both:

```
"contact-endpoint=<label>"
```

```
"contact-firewall=<label>"
```

```
"contact-endpoint=<label>, contact_firewall=<label>""
```

6. Press <Enter>.

For example, if you want your Net-Net SBC to add private endpoint and public firewall information to Contact headers, and you want to label this information as **ep** and **fw**, you would enter the following information in the ACLI.

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

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

```
ACMEPACKET(sip-config)# options "contact-
endpoint="ep", contact_firewall="fw""
```

## Embedded Header Support

---

This section explains how to configure embedded header support. The Net-Net SBC 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 SBC. 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=%3Csp:+1-ESQK@IPv4_My_EAG; user=phone%3E>
```

Then, if you configure your Net-Net SBC 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            real m=            ip=1.0.0.0  
2: SIPSA        real m=            ip=10.10.102.1  
  
selection: 2  
ACMEPACKET(session-agent)#
  5. **request-uri-headers**—Enter a list of embedded headers extracted from the Contact header that will be inserted in the re INVITE message. To configure this parameter for multiple headers, enclose the headers in double quotes and separate them with spaces. This completes the configuration of embedded header support.  
ACMEPACKET(session-agent)# **request-uri-headers P-Asserted-Identity**

## Static SIP Header and Parameter Manipulation

---

This section explains the SIP header and parameter manipulation feature, which lets the Net-Net SBC add, modify, and delete SIP headers and parts of SIP headers called SIP header elements. SIP header elements are the different subparts of the header, such as the header value, header parameter, URI parameter and so on (excluding the header name).

To enable the SIP header and parameter manipulation functionality, you create header manipulation rulesets in which you specify header manipulation rules, as well as optional header element rules that operate on specified header elements. You then apply the header manipulation ruleset as inbound or outbound for a session agent or SIP interface.

### Header Manipulation Rules

Header manipulation rules operate on the header you specify when you configure the rule. A header manipulation rule can also be configured with a list of element rules, each of which would specify the actions you want performed for a given element of this header.

### Header Element Rules

Header element rules perform operations on the elements of a header. Header elements include all subparts of a header; excluding the header name. For example, header value, header parameter, URI parameter, and so on.

## About SIP Header and Parameter Manipulation

Using the SIP header manipulation ruleset, you can cause the Net-Net SBC 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 SBC 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 SBC where and how to manipulate the SIP message to use originating (access) and terminating (egress) trunk group URI parameters.

See [Trunk Group URIs \(406\)](#) for more information about trunk group URIs.

## ACLI Instructions and Examples

This section explains how to configure SIP header and parameter manipulation. First you create a SIP header manipulation ruleset, then the header manipulation rules and optional header element rules you want that ruleset to contain. You then configure a session agent or a SIP interface to use the SIP header and parameter manipulation ruleset in the inbound and outbound directions.

### Creating SIP Header Manipulation Rulesets

#### To configure the SIP header manipulation ruleset:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-router path.   
ACMEPACKET(config)# **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(sessi on-router)# si p-mani pul ati on
ACMEPACKET(si p-mani pul ati on)#

```

4. **name**—Enter the name you want to use for this ruleset.
5. **header-rules**—Define the header manipulation rules you want to include in this ruleset.

5a. Type **header-rules** and press <Enter>.

```
ACMEPACKET(si p-mani pul ati on)# header-rul es
ACMEPACKET(si p-header-rul es)#

```

- 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. The default value is **none**. Valid options are:

- **add**—Add a new header, if that header does not already exist.
- **delete**—Delete the header, if it exists.
- **manipulate**—Elements of this header will be manipulated according to the element rules configured.
- **store**—Store the header.
- **none**—No action to be taken.

- 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. The default value is **any**. Valid options are:

- **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(si p-header-rul es)# el ement-rul es
ACMEPACKET(si p-el ement-rul es)#

```

- 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. The default value is **none**. Valid options are:

- **header-value**—Enter value of the header.

- **header-param-name**—Header parameter name.
  - **header-param**—Parameter portion of the header.
  - **uri-display**—Display of the SIP URI.
  - **uri-user**—User portion of the SIP URI.
  - **uri-host**—Host portion of the SIP URI.
  - **uri-port**—Port number portion of the SIP URI.
  - **uri-param-name**—Name of the SIP URI param.
  - **uri-param**—Parameter included in the SIP URI.
  - **uri-header-name**—SIP URI header name
  - **uri-header**—Header included in a request constructed from the URI.
  - **uri-user-param**—User parameter of the SIP URI.
- 6d. **action**—Enter the action you want applied to the element specified in the name parameter, if there is a match value. The default value is **none**. Valid options are:
- **none**—No action is taken.
  - **add**—Add a new element, if it does not already exist.
  - **replace**—Replace the elements.
  - **delete-element**—Delete the specified element if it exists. Based on the match value if entered in step 6f.
  - **delete-header**—Delete the specified header, if it exists.
  - **store**—Store the elements.
- 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. The default value is **ANY**. Valid options are:
- **IP**—Element value in the SIP message must be a valid IP address to be compared to the match-value field entry. If the match-value field is empty, any valid IP address is considered a match. If the element value is not a valid IP address, it is not considered a match.
  - **FQDN**—Element value in the SIP message must be a valid FQDN to be compared to the match-value field entry. If the match-value field is empty, any valid FQDN is considered a match. If the element value is not a valid FQDN, it is not considered a match.
  - **ANY**—Element value in the SIP message is compared with the match-value field entry. If the match-value field is empty, all values are considered a match.
- 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 a new element or to replace a value for an existing element. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators.
- Absolute values, with which you can use double quotes for clarity. You must escape all double quotes and back slashes that are part of an absolute value, and enclose the absolute value in double quotes.

For example:

```
"si p: "+$TRUNK_GROUP+" . \"$TRUNK_GROUP_CONTEXT
```

- Pre-defined parameters always start with a \$. Valid pre-defined parameters are:

| Parameter             | Description                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| \$ORIGINAL            | Original value of the element is used.                                                                                            |
| \$LOCAL_IP            | IP address of the SIP interface on which the message was received for inbound manipulation; or sent on for outbound manipulation. |
| \$REMOTE_IP           | IP address the message was received from for inbound manipulation; or being sent to for outbound manipulation.                    |
| \$REMOTE_VIA_HOST     | Host from the top Via header of the message is used.                                                                              |
| \$TRUNK_GROUP         | Trunk group is used.                                                                                                              |
| \$TRUNK_GROUP_CONTEXT | Trunk group context is used.                                                                                                      |

| Operator | Description                                                                             |
|----------|-----------------------------------------------------------------------------------------|
| +        | Append the value to the end. For example:<br>"acme"+ "packet"<br>generates "acmepacket" |
| +^       | Prepends the value. For example:<br>"acme"+^ "packet"<br>generates "packetacme"         |
| -        | Subtract at the end. For example:<br>"112311"- "11"<br>generates "1123"                 |
| -^       | Subtract at the beginning. For example:<br>"112311"-^ "11"<br>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:  
**I INVITE, ACK, BYE**  
This field is empty by default. If you leave the method field empty, the header-rule is applied to all methods.
8. Type **exit** to return to the sip-manipulation level.

9. Save your work using the ACLI **done** command.
10. If you want to save this configuration, exit out of configuration mode and type **save-config**.  
See the next section for examples of SIP header manipulation ruleset configurations.

## Configuring a Session Agent

You can configure a session agent to use the SIP header manipulation ruleset.

### To configure a session agent:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-router path.   
ACMEPACKET(configure)# **session-router**
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(session-router)# **session-agent**  
ACMEPACKET(session-agent)#
4. **in-manipulationid**—Enter the name of the SIP header manipulation ruleset you want to apply to inbound SIP packets.   
ACMEPACKET(session-agent)# **in-manipulationid route-stripper**
5. **out-manipulationid**—Enter the name of the SIP header manipulation ruleset you want to apply to outbound SIP packets.   
ACMEPACKET(session-agent)# **out-manipulationid route-stripper**
6. Save your work using the ACLI **done** command.
7. If you want to save this configuration, exit out of configuration mode and type **save-config**.

## Configuring a SIP Interface

You can configure a interface to use a SIP header manipulation ruleset.

### To configure a SIP interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-router path.   
ACMEPACKET(configure)# **session-router**
3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(session-router)# **sip-interface**  
ACMEPACKET(sip-interface)#
4. **in-manipulationid**—Enter the name of the SIP header manipulation ruleset you want to apply to SIP packets in the ingress direction.   
ACMEPACKET(sip-interface)# **in-manipulationid**
5. **out-manipulationid**—Enter the name of the SIP header manipulation ruleset you want to apply to SIP packets in the egress direction.   
ACMEPACKET(sip-interface)# **out-manipulationid**
6. Save your work using the ACLI **done** command.

7. If you want to save this configuration, exit out of configuration mode and type **save-config**.

### **Example 1: Stripping All Route Headers**

This example explains how to strip all route headers from a SIP packet. First, you create a header manipulation ruleset, in the example it is called **route-stripper**. Then you configure the list of header manipulation rules you need to strip route headers. In this case, you only need one rule named **Route** (to match the **Route** header name) with the action set to **Delete**.

```
ACMEPACKET# config terminal
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)# name route-stripper
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)# name Route
ACMEPACKET(sip-header-rules)# action Delete
ACMEPACKET(sip-header-rules)# done
header-rule
 name Route
 action delete
 match-value
 msg-type any
ACMEPACKET(sip-header-rules)# ex
ACMEPACKET(sip-manipulation)# done
sip-manipulation
 name route-stripper
 header-rule
 name Route
 action delete
 match-value
 msg-type any
```

### **Example 2: Stripping an Existing Parameter and Adding a New One**

This example explains how to strip the user parameter from the Contact header URI and add the acme parameter with value as LOCAL IP, only for requests. First you create a header manipulation ruleset, in the example it is called **param-stripper1**. You then configure a list of header rules you need. In this case, you only need one rule named **Contact** (to match the **Contact** header name), with action set to **manipulate** (indicating the elements of this header would be manipulated). Next, you configure a list of element rules for the **Contact** header rule.

In this case you configure two element rules; one to strip the **uri** parameter **user** (the rule name **user** matches the param name **user**) and the other to add the **uri** parameter **acme** (the rule name **acme** matches the param name **acme**).

```
ACMEPACKET# config terminal
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)# sip-manipulation
ACMEPACKET(sip-manipulation)# name param-stripper1
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)# name Contact
```

```

ACMEPACKET(sip-header-rules)# action manipulate
ACMEPACKET(sip-header-rules)# msg-type request
ACMEPACKET(sip-header-rules)# element-rules
ACMEPACKET(sip-element-rules)# name user
ACMEPACKET(sip-element-rules)# type uri-param
ACMEPACKET(sip-element-rules)# action delete-element
ACMEPACKET(sip-element-rules)# done
element-rule
 name user
 type uri-param
 action delete-element
 match-val-type any
 match-value
 new-value

ACMEPACKET(sip-element-rules)# name acme
ACMEPACKET(sip-element-rules)# action add
ACMEPACKET(sip-element-rules)# type uri-param
ACMEPACKET(sip-element-rules)# new-value "$LOCAL_IP"
ACMEPACKET(sip-element-rules)# done
element-rule
 name acme
 type uri-param
 action add
 match-val-type any
 match-value
 new-value "$LOCAL_IP"

ACMEPACKET(sip-element-rules)# ex
ACMEPACKET(sip-header-rules)# done
header-rule
 name Contact
 action manipulate
 match-value
 msg-type request
element-rule
 name user
 type uri-param
 action delete-element
 match-val-type any
 match-value
 new-value

element-rule
 name acme
 type uri-param
 action add
 match-val-type any
 match-value

```

```

new-value "$LOCAL_IP"
ACMEPACKET(sip-header-rules)# ex
ACMEPACKET(sip-manipulation)# done
sip-manipulation
 name param-stripper1
 header-rule
 name Contact
 action manipulate
 match-value
 msg-type request
 element-rule
 name user
 type uri-param
 action delete-element
 match-val-type any
 match-value
 new-value
 element-rule
 name acme
 type uri-param
 action add
 match-val-type any
 match-value
 new-value "$LOCAL_IP"

```

For example, if the IP address of the SIP interface (\$LOCAL\_IP) is 10.1.2.3 and the Net-Net SBC receives the following Contact header:

Contact: <sip:1234@10.4.5.6; user=phone>

The header rule is applied to strip the user parameter from the Contact header URI and add the acme parameter with the value 10.1.2.3:

Contact: <sip:1234@10.4.5.6; acme=10.1.2.3>

## SIP HMR (Header Manipulation Rules)

---

SIP header manipulation can also be configured in a way that makes it possible to manipulate the headers in SIP messages both statically and dynamically. Using this feature, you can edit response headers or the Request-URI in a request, and change the status code or reason phrase in SIP responses.

Available in Net-Net OS Release 4.0 and later, [Static SIP Header and Parameter Manipulation \(315\)](#) allows you to set up rules in your Net-Net SBC configuration that remove and/or replace designated portions of specified SIP headers. SIP HMR allows you to set up dynamic header manipulation rules, meaning that the Net-Net SBC has complete control over alterations to the header value. More specifically:

- The Net-Net SBC can search header for dynamic content or patterns with the header value. It can search, for example, for all User parts of a URI that begin with 617 and end with 5555 (e.g., 617...5555).

- The Net-Net SBC can manipulate any part of a patterns match with any part of a SIP header. For example, 617 123 5555 can become 617 231 5555 or 508 123 0000, or any combination of those.

To provide dynamic header manipulation, the 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 change what you want to change.

## 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. 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 takes 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 second rule's match value finds a hit (and only 10 matches, 0-9, are permitted) for the results (yield) from applying the first rule.

Consider the example of the following regular expression entry made for a **match-value** parameter: 'Trunk(.+)', which might be set as that match value in the first rule you configure. Given a SIP element rule called `uri-param` and the param-name `tgid`, it can yield two values:

- Grouping 0—The entire matching string (`Trunk1`)
- Grouping 1—The grouping (1)

In turn, these groupings can be referenced in an element rule by using this syntax:

```
$<header rule name>. $<element rule name>. $<value>
```

Additional syntax options that can be used with this feature are:

- `$headerName['[' index']']`
- `$headerName['[' index']'][. $index]`
- `$headerName['[' index']'][. $elementName]`
- `$headerName['[' index']'][. $elementName][. $index]`

## Guidelines for Header and Element Rules

Header rules and element rules share these guidelines:

- References to groupings that do not exist result in an empty string.
- References to element rule names alone result in a Boolean condition of whether the expression matched or not.
- A maximum of ten matches are allowed for a regular expression. Match 0 (grouping 0) is always the match of the entire matching string; subsequent numbers are the results for other groups that match.

## Precedence

The 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.

This ordering also provides you with ways to use manipulations strategically. For example, you might want to use two rules if you want to store the values of a regular expression. The first rule would store the value of a matched regular expression, and the second could delete the matched value.

In addition to taking note of the order in which header rules are configured, you now must also configure a given header rule prior to referencing it. For example, you must create Rule1 with the action store for the Contact header BEFORE you can create Rule2 which uses the stored value from the Contact header.

## Duplicate Header Names

If more than one header exists for the header name you have configured, the 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.

## 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 header rules configuration now takes the value `si p-mani p`. When you set the parameter to `si p-mani p`, 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

## MIME Support

Using the SIP HMR feature set, you can manipulate MIME types in SIP message bodies. While you can manipulate the body of SIP messages or a specific content type using other iterations of SIP HMR, this version gives you the power to change the MIME attachment of a specific type within the body by using regular expressions. To achieve this, you use the **find-replace-all** action type, which enables the search for a particular string and the replacement of all matches for that type. Although you use **find-replace-all** to manipulate MIME attachments, it can also be used to achieve other goals in SIP HMR.

Note that using **find-replace-all** might consume more system resources than other HMR types. Therefore this powerful action type should only be used when another type cannot perform the type of manipulation you require.

## 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.

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 (abbreviated as c).

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., appl i cati on/\*). 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 <b>new-value</b> you configure.                                                   |
| find-replace-all | Searches the specifies mime-type's contents and replaces all matching regular expressions with the <b>new-value</b> 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: appl i cati on/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=audi o 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 [ . . . ]

## SIP Message-Body Separator Normalization

The Net-Net SBC supports SIP with Multipurpose Internet Mail Extension (MIME) attachments — up to a maximum payload size of 64KB — and has the ability to allow more than the required two CRLFs between the SIP message headers and the multipart body's first boundary. The first two CRLFs that appear in all SIP messages signify the end of the SIP header and the separation of the header and body of the message, respectively. Sometimes additional extraneous CRLFs can appear within the preamble before any text.

The Net-Net SBC works by forwarding received SIP messages regardless of whether they contain two or more CRLFs. Although three or more CRLFs are legal, some SIP devices do not accept more than two.

The solution to ensuring all SIP devices accept messages sent from the Net-Net SBC is to strip all CRLFs located at the beginning of the preamble before the appearance of any text, ensuring that there are no more than two CRLFs between the end of the last header and the beginning of the body within a SIP message. You enable this feature by adding the new `stripPreambleCrlf` option to the global SIP configuration.

### To enable the stripping of CRLFs in the preamble:

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

```
ACMEPACKET# config terminal
ACMEPACKET(configure)#
```
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**—Set the options parameter by typing `options`, a <Space>, the option name `stripPreambleCrlf` with a “plus” sign in front of it, and then press <Enter>.  

```
ACMEPACKET(sip-config)# options +stripPreambleCrlf
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the global SIP configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

## 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.
- Do header rule name that is all capital letters; there are predefined rules that are used as macros, and they might conflict with a name that uses capital letters; e.g., \$I P\_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.                                                                                   |
| *         | star/asterisk         | 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.                                                              |
| *         | star/asterisk         | Typically used with a . (dot) in the format . * to indicate that a match for any character, 0 or more times.                                                                                           |
| *         | star/asterisk         | Literally an * (asterisk) when bracketed ([ ]).                                                                                                                                                        |
| +         | plus                  | Matches one or more of the preceding character, bracketed carrier class, or group in parentheses. Used for quantification.                                                                             |
| +         | plus                  | 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.<br><br>Interval ranges are entered as minimum and maximums ({mi ni mum, maxi mum}) 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 ({mi ni mum, }).                                                                      |
| ?         | 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.<br><br>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: <ul style="list-style-type: none"> <li>• They separate pattern strings so that a whole string can have special characters within it as if it were a single character.</li> <li>• They allow the designated pattern to be stored and referenced later (so that other operations can be performed on it).</li> </ul>                                                                                                                                                                                        |

## Expression Building Using Parentheses

You can now use parentheses ( ) when you use HMR to support order of operations and to simplify header manipulation rules that might otherwise prove complex. This means that expressions such as "(si p + urp) - (u + rp)" can now be evaluated to si p. Previously, the same expression would have evaluated to si purprp. In addition, you previously would have been required to create several different manipulation rules to perform the same expression.

## ACLI Instructions and Examples

### Configuring SIP Header Manipulation Rules

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.

#### 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(config)# 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.  
Set this parameter to @status-line, where the at-sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name "status-code."
7. **msg-type**—Specify the type of message to which this SIP HMR will be applied. The default value is **any**. The valid values are:
  - any | request | reply
8. **methods**—Enter the method type to use when this SIP HMR is used, such as INVITE, ACK, or CANCEL. When you do not set the method, the 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. This choice dictates how the Net-Net SBC processes the match rules against the SIP header. the default is **refer-case-sensitive**. The valid values are:
  - boolean | refer-case-sensitive | refer-case-insensitive | pattern-rule
10. **action**—Enter the action that you want this rule to perform on the SIP header. The default value is **none**. The valid values are:
  - add | delete | manipulate | store | none

Remember that you should enter rules with the action type **store** before you enter rules with other types of actions.

When you set the action type to **store**, the Net-Net SBC always treats the match value you enter as a regular expression. As a default, the regular expression is uses for the match value is .+ (which indicates a match value of at least one character), unless you set a more specific regular expression match value.

11. **match-value**—Enter the value to match against the header value in SIP packets; the 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.

When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators..

You can also escape the escape character itself, so that it is used as a literal string. For example, the Net-Net SBC now treats the string \+1234 as +1234.

The following are escape characters: +, -, +^, -^, &, |, \, (, ), ., \$, ^, and “.”.

You can also use two variables, \$REMOTE\_PORT and \$LOCAL\_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

12. **new-value**—When the action parameter is set to add or to manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Net-Net SBC to use when it adds or manipulates SIP headers.

When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the **new-value** parameter to support escaping Boolean and string manipulation operators..

You can also escape the escape character itself, so that it is used as a literal string. For example, the Net-Net SBC now treats the string \+1234 as +1234.

The following are escape characters: +, -, +^, -^, &, |, \, (, ), ., \$, ^, and “.”.

You can also use two variables, \$REMOTE\_PORT and \$LOCAL\_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

## Configuring SIP Header Manipulation Element Rules

Element rules are a subset of the SIP header manipulation rules and are applied at the element type level rather than at the entire header value.

### To configure dynamic SIP header manipulation rules:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.  
ACMEPACKET(configure)# **session-router**  
ACMEPACKET(session-router)#
 

```
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. There is no default for this parameter.
```

8. **type**—Specify the type of parameter to which this element rule will be applied. The default value is **none**. The valid values are:

- header-value | header-param-name | header-param | uri-display | uri-user | uri-user-param | uri-host | uri-port | uri-param-name | uri-param | uri-header-name | uri-header

To configure HMR so that there is impact only on the status-line; the value will be used for matching according to the **comparison-type**:

- **status-code**—Designates the status code of the response line; accepts any string, but during the manipulation process only recognizes the range from 100 to 699.
- **reason-phrase**—Designates the reason of the response line; accepts any string.

9. **match-val-type**—Enter the value type that you want to match when this rule is applied. The default value is **ANY**. Valid values are:

- IP | FQDN | ANY

10. **comparison-type**—Enter the way that you want SIP headers to be compared from one of the available. This choice dictates how the Net-Net SBC processes the match rules against the SIP header parameter/element. The default is **refer-case-sensitive**.

- boolean | refer-case-sensitive | refer-case-insensitive | pattern-rule

11. **action**—Enter the action that you want this rule to perform on the SIP header parameter/element. The default is **none**. The valid rules are:

- add | replace | delete-element | delete-header | store | none

Remember that you should enter rules with the action type **store** before you enter rules with other types of actions.

When you set the action type to **store**, the Net-Net SBC always treats the match value you enter as a regular expression. As a default, the regular expression is uses for the match value is **.+** (which indicates a match value of at least one character), unless you set a more specific regular expression match value.

12. **match-value**—Enter the value to match against the header value in SIP packets; the 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.

When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators..

You can also escape the escape character itself, so that it is used as a literal string. For example, the Net-Net SBC now treats the string **\+1234** as **+1234**.

The following are escape characters: **+, -, +^, -^, &, |, \, (, ), ., \$, ^, and “.**

You can also use two variables, **\$REMOTE\_PORT** and **\$LOCAL\_PORT**, which resolve respectively to the far-end and remote UDP or TCP port value.

13. **new-value**—When the action parameter is set to **add** or to **manipulate**, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Net-Net SBC to use when it adds or manipulates parameters/elements.

When you configure HMR (using SIP manipulation rules, elements rules, etc.), you can now use escape characters in the **new-value** parameter to support escaping Boolean and string manipulation operators..

You can also escape the escape character itself, so that it is used as a literal string. For example, the Net-Net SBC now treats the string \+1234 as +1234.

The following are escape characters: +, -, +^, -^, &, |, \, (, ), ., \$, ^, and “.”.

You can also use two variables, \$REMOTE\_PORT and \$LOCAL\_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

## **Status-Line Manipulation and Value Matching**

The Net-Net SD’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 SD to drop a 183 Session Progress response when it does not have SDP, though flexibility is built into this feature so that you can use it to achieve other ends. In addition, you can now set the SIP manipulation’s **match-value** parameter with Boolean parameters (AND or OR).

## **Setting the Header Name**

SIP header rules (part of the SIP manipulation configuration) now support a new value for the header-name parameter. The value is @status-line, where the at-sign (@)—not allowed in SIP header names—prevents undesired matches with header having the name “status-code.”

### **To set the header name for SIP header rules:**

1. In Superuser mode, type **configure terminal** and press <Enter>.   
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.   
ACMEPACKET(configure)# **session-router**  
ACMEPACKET(session-router)#
3. Type **sip-manipulation** and press <Enter>.   
ACMEPACKET(session-router)# **sip-manipulation**  
ACMEPACKET(sip-manipulation)#
4. Type **header-rules** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(session-router)# **header-rules**  
ACMEPACKET(sip-header-rules)#
5. **header-name**—Enter the new value for the **header-name** parameter: @status-line.

## **Setting the Element Type**

In the element rules (a subset of the SIP header rules configuration), you can now set the **type** parameter to either of the following values, both of which will only have an impact on the status-line:

- **status-code**—Designates the status code of the response line; accepts any string, but during the manipulation process only recognizes the range from 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 SD 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# config 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 SD 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 SD 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# config 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 SD 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 SD matches these against the value of the parameter/element. This is where you can enter values to match using regular expression values, or stored pattern matches; your entries can contain Boolean operators.

## Setting the Response Code Block

To enable the SIP HMR enhancements, you need to set an option in SIP interface configuration that keeps the Net-Net SD from sending the response you designate.

Note that this example sets the dropResponse option to 699, where 699 is an arbitrary code used to later match the HMR.

**To enable SIP response blocking for a SIP interface:**

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

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

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

3. Type **sip-interface** and press <Enter>.

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

```
ACMEPACKET(sip-interface) #
```

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

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **dropResponse** with a “plus” sign in front of it, type the equal sign and the code(s) or range(s) you want blocked. If there is more than one, separate your entries with a colon. Then press <Enter>.

```
ACMEPACKET(si p-interface)# options +dropResponse=699
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

## Configuring MIME Support

The **find-replace-all** action has been added to the header rules. Element rules support the **find-replace-all** action and the **mime** type.

### To set the header rule with the find-replace-all action:

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

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

```

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

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

```

3. Type **sip-manipulation** and press <Enter>.

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

```

4. Type **header-rules** and press <Enter>.

```
ACMEPACKET(sip-manipulation)# header-rules
```

5. ACMEPACKET(sip-header-rules)#

6. **action**—Set the action parameter to **find-replace-all** if you want to enable SIP HMR MIME manipulation.

7. Save and activate your configuration.

### To set the element rule with the find-replace-all action and MIME type:

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

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

```

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

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

```

3. Type **sip-manipulation** and press <Enter>.

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

```

4. Type **header-rules** and press <Enter>.

- ```
ACMEPACKET(sip-manipulation)# header-rules
ACMEPACKET(sip-header-rules)#
5. Type element-rules and press <Enter>
   ACMEPACKET(sip-header-rules)# element-rules
6. ACMEPACKET(sip-element-rules)#
7. action—Set the action parameter to find-replace-all if you want to enable SIP HMR MIME manipulation.
8. type—Set the type parameter to mime if you want to enable SIP HMR MIME manipulation.
9. Save and activate your configuration.
```

Testing Pattern Rules

The Net-Net SBC supports a new command that allows you to test the regular expression that you might use in SIP manipulation rules to see if it yields the results you require. This command is useful for testing the regex values that you devise because it will tell you whether that value is valid or not.

This new command is called **test-pattern-rule**, and you can access it through the ACLI's session-router path.

To test a pattern 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-level configuration elements.


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

```
3. Type **test-pattern-rule** and press <Enter>.


```
ACMEPACKET(session-router)# test-pattern-rule
ACMEPACKET(test-pattern-rule)#

```
4. **expression**—Enter the regular expression that you want to test. If there is a match, then the Net-Net SBC will inform you of it; you will also be informed if there is no match.

The “string” against which the Net-Net SBC is matching is not the string parameter that you can use for this command; it is the string value of the regular expression you entered.

```
ACMEPACKET(test-pattern-rule)# expression '. *; tgid=(.+) . *'
```

5. **string**—Enter the string against which you want to compare the regular expression.

```
ACMEPACKET(test-pattern-rule)# string
sip: +17024260002@KCMGGWC; user=phone SIP/2.0; tgid=Trunk1
expression made 3 matches against string
```

6. **show**—Use the **show** command within **test-pattern-rules** to view the test pattern that you entered, whether there was a match, and the number of matches.

```
ACMEPACKET(test-pattern-rule)# show
Pattern Rule:
  Expression : . *(; tgid=(.+)). *
  String     : sip: +17024260002@KCMGGWC; user=phone SIP/2.0; tgid=Trunk1
  Matched    : TRUE
```

Matches:

```
$0 sip: +17024260002@KCMGGWC; user=phone SIP/2.0; tgi d=Trunk1
$1 ; tgi d=Trunk1
$2 Trunk1
```

Configuring SIP HMR Sets

This section shows you how to configure your multiple SIP HMR sets. Refer to [Example 10: Use of SIP HMR Sets \(355\)](#) for a sample configuration.

Remember to run the ACLI **verify-config** command prior to activating your configuration so the Net-Net SBC can detect any invalid or circular references.

To set the parameters enabling the use of SIP HMR sets:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **configure terminal**
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.
 ACMEPACKET(configure)# **session-router**
 ACMEPACKET(session-router)#
3. Type **sip-manipulation** and press <Enter>.
 ACMEPACKET(session-router)# **sip-manipulation**
 ACMEPACKET(sip-manipulation)#
4. Type **header-rules** and press <Enter>.
 ACMEPACKET(session-router)# **header-rules**
 ACMEPACKET(sip-header-rules)#
5. Type **element-rule** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
 ACMEPACKET(sip-header-rules)# **element-rules**
 ACMEPACKET(sip-element-rules)#
6. **action**—Enter the **sip-manipulation** value to enable use this rule for a SIP HMR set. This value then invoke the rule identified in the new-value parameter.
7. **new-value**—To use SIP HMR sets, enter the name of the manipulation rule you want invoked for the set.
8. Save and activate your configuration.

Configuration Examples

This section shows you several configuration examples for HMR. This section shows the configuration for the various rules that the 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		removeHeader
header-rule		
name		removeCustom

header-name	Custom
action	delete
comparison-type	boolean
match-value	^This is my.*
msg-type	request
new-value	
methods	INVITE
header-rule	
name	goodByeHeader
header-name	Goodbye
action	store
comparison-type	boolean
match-value	^Remove (.+)
msg-type	request
new-value	
methods	INVITE
header-rule	
name	goodBye
action	delete
comparison-type	pattern-rule
match-value	\$goodByeHeader
msg-type	request
new-value	
methods	INVITE

This is a sample of the result:

```
Request-Line: INVITE sip:servi ce@192.168.200.60:5060; tgi d=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060; branch=z9hG4bK0g639r10fgc0akk26s1.1
From: sipp <sip:sipp@192.168.1.60:5060>; tag=Sdc1rm601-1
To: sut <sip:servi ce@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	ruri Regex
header-name	request-uri
action	store
comparison-type	pattern-rule
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	tgidParam
parameter-name	tgid
type	uri-param
action	store
match-value-type	any
comparison-type	pattern-rule
match-value	
new-value	
header-rule	
name	addCustomer
header-name	X-Customer-Profile
action	add
comparison-type	pattern-rule
match-value	\$ruri Regex. \$tgidParam
msg-type	request
new-value	\$ruri Regex. \$tgidParam. \$0
methods	INVITE
header-rule	
name	delTgid
header-name	request-uri
action	manipulate
comparison-type	pattern-rule
match-value	\$ruri Regex. \$tgidParam
msg-type	request
new-value	
methods	INVITE
element-rule	
name	tgidParam
parameter-name	tgid
type	uri-param
action	delete-element
match-value-type	any
comparison-type	case-sensitive
match-value	\$ruri Regex. \$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: sip <sip:sip@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: <si p: si pp@192. 168. 200. 61: 5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: si pp <si p: user@192. 168. 1. 60: 5060; up=abc>; hp=123
Params: si pp <si p: si pp1@192. 168. 1. 60: 5060>
Params: si pp <si p: si pp2@192. 168. 1. 60: 5060>
Edit: di sp <si p: 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 si p: service@192. 168. 200. 60: 5060; tgid=123 SIP/2. 0
Message-Header
Via: SIP/2. 0/UDP
192. 168. 200. 61: 5060; branch=z9hG4bK20q2s820boghbacgs6o0. 1

```

```

From: <sip:si pp@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:si pp@192.168.200.61:5060; transport=udp>
Goodbye: Remove Me
Custom: This is my edited header
Display: <sip:sip:user@192.168.1.60:5060; up=abc>; hp=123
Params: <sip:si pp1@192.168.1.60:5060>
Params: <sip:si pp2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

```

Example 4: Storing and Using URI Parameters

For this manipulation rule, the 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	storeElementParam	
header-rule		
name	Frohmr	
header-name	From	
action	store	
comparison-type	case-sensitive	
match-value		
msg-type	request	
new-value		
methods	INVITE	
element-rule		
name	elementRule	
parameter-name	tag	
type	uri-param	
action	store	
match-val-type	any	
comparison-type	case-sensitive	
match-value		
new-value		
header-rule		
name	newHeader	
header-name	FromTag	
action	add	
comparison-type	pattern-rule	
match-value	\$FromHR.\$elementRule	
msg-type	any	
new-value	\$FromHR.\$elementRule.\$0	
methods		

This is a sample of the result:

```

Request-Line: INVITE sipp: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<sipp:service@192.168.1.60:5060>; tag=SDF1re601-1
To: sut <sipp:service@192.168.1.61:5060>
Call-ID: SDF1re601-f85059e74e1b443499587dd2dee504c2-06a3gu0
CSeq: 1 INVITE
Contact: <sipp:sipp@192.168.200.61:5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sipp<sipp:user@192.168.1.60:5060; up=abc>; hp=123
Params: sipp<sipp:sipp1@192.168.1.60:5060>
Params: sipp<sipp:sipp2@192.168.1.60:5060>
Edit: disp<sipp:user@192.168.1.60:5060>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
From-Tag: 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 "abc123" if it matches "sipp."

This is a sample of the configuration:

<pre> sipp-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 pattern-rule match-value new-value header-rule name header-name action </pre>	<pre> modDisplayParam storeDisplay Di spl ay store case-sensitive request INVITE di spl ayName di spl ay uri-di spl ay store any (s)(i p)(p) modDisplay Di spl ay manipulate </pre>
---	--

compari son-type	case-sensi ti ve
match-val ue	
msg-type	request
new-val ue	
methods	I NVI TE
el ement-rul e	
name	modRul e
parameter-name	di spl ay
type	uri -di spl ay
acti on	repl ace
match-val -type	any
compari son-type	pattern-rul e
match-val ue	
\$storeDi spl ay. \$di spl ayName	
new-val ue	
\$storeDi spl ay. \$di spl ayName. \$1+!ur+\$storeDi spl ay. \$di spl ayName. \$3	
header-rul e	
name	modFrom
header-name	From
acti on	mani pul ate
compari son-type	pattern-rul e
match-val ue	
msg-type	request
new-val ue	
methods	I NVI TE
el ement-rul e	
name	fromDi spl ay
parameter-name	
type	uri -di spl ay
acti on	repl ace
match-val -type	any
compari son-type	pattern-rul e
match-val ue	si pp
new-val ue	"\"abc_123\" "

This is a sample of the result:

```

Request-Line: I NVI TE si p: servi ce@192. 168. 200. 60: 5060; tgi d=123 SIP/2. 0
Message Header
Via: SIP/2. 0/UDP
192. 168. 200. 61: 5060; branch=z9hG4bK681kot109gp04acgs6o0. 1
From: "abc_123" <si p: si pp@192. 168. 1. 60: 5060>; tag=SD79ra601-1
To: sut <si p: servi ce@192. 168. 1. 61: 5060>
Call-ID: SD79ra601-a487f1259e2370d3dbb558c742d3f8c4-06a3gu0
CSeq: 1 I NVI TE
Contact: <si p: si pp@192. 168. 200. 61: 5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: slurp <si p: user@192. 168. 1. 60: 5060; up=abc>; hp=123
Params: si pp <si p: si pp1@192. 168. 1. 60: 5060>
Params: si pp <si p: si pp2@192. 168. 1. 60: 5060>
Edit: disp <si p: 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	elementParams
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	
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-val ue	
el ement-rul e		
	name	headerParam
	parameter-name	hp
	type	header-param
	acti on	store
	match-val -type	any
	compari son-type	pattern-rul e
	match-val ue	
	new-val ue	
header-rul e		
	name	EditParams
	header-name	Params
	acti on	manipulate
	compari son-type	case-sensi ti ve
	match-val ue	
	msg-type	request
	new-val ue	
	methods	I N V I T E
el ement-rul e		
	name	addHeaderParam
	parameter-name	display
	type	header-param
	acti on	add
match-val -type	any	
	compari son-type	case-sensi ti ve
	match-val ue	
	new-val ue	
\$StoreDi spl ay. \$di spl ayName. \$0		
el ement-rul e		
	name	addUriParam
	parameter-name	user
	type	uri-param
	acti on	add
	match-val -type	any
	compari son-type	case-sensi ti ve
	match-val ue	
	new-val ue	
\$StoreDi spl ay. \$userNa me. \$0		
header-rul e		
	name	EditDi spl ay
	header-name	Di spl ay
	acti on	manipulate
	compari son-type	case-sensi ti ve
	match-val ue	
	msg-type	request
	new-val ue	
	methods	I N V I T E
el ement-rul e		
	name	replaceUriParam
	parameter-name	up
	type	uri-param
	acti on	replace
	match-val -type	any

compari son-type	pattern-rule
match-val ue	\$StoreDi spl ay. \$uri Param
new-val ue	def
el ement-rul e	
name	del HeaderParam
parameter-name	hp
type	header-param
acti on	del ete-el ement
match-val -type	any
compari son-type	pattern-rule
match-val ue	\$StoreDi spl ay. \$headerParam
new-val ue	

This is a sample of the result:

```

Request-Line: INVITE si p: service@192.168.200.60:5060; tgi d=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060; branch=z9hG4bK7okvei 0028J gdacgh6c1.1
From: sipp <si p: sipp@192.168.1.60:5060>; tag=SD89rm601-1
To: sut <si p: service@192.168.1.61:5060>
Call-ID: SD89rm601-b5b746cef19d0154cb1f342cb04ec3cb-06a3gu0
CSeq: 1 INVITE
Contact: <si p: sipp@192.168.200.61:5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sipp <si p: user@192.168.1.60:5060; up=def>
Params: sipp
<si p: sipp@192.168.1.60:5060; user=user>; display=sipp
Params: sipp
<si p: sipp@192.168.1.60:5060; user=user>; display=sipp
Edit: disp <si p: 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:

si p-mani pul ati on	
name	Params
header-rul e	
name	storeParams
header-name	Params
acti on	store
compari son-type	case-sensi ti ve
match-val ue	
msg-type	request
new-val ue	
methods	INVITE
el ement-rul e	

	name	storeUserName
	parameter-name	user
	type	uri-user
	action	store
	match-val-type	any
	comparison-type	case-sensitive
	match-value	
	new-value	
header-rule		
	name	modEdit
	header-name	Edit
	action	manipulate
	comparison-type	pattern-rule
	match-value	
	msg-type	request
	new-value	
methods	I INVITE	
	element-rule	
	name	addParam1
	parameter-name	c1
	type	uri-param
	action	add
	match-val-type	any
	comparison-type	case-sensitive
	match-value	
	new-value	
\$storeParams[0].\$storeUserName.\$0	element-rule	
	name	addParam2
	parameter-name	c2
	type	uri-param
	action	add
	match-val-type	any
	comparison-type	case-sensitive
	match-value	
	new-value	
\$storeParams[1].\$storeUserName.\$0	element-rule	
	name	addParam2
	parameter-name	c2
	type	uri-param
	action	add
	match-val-type	any
	comparison-type	case-sensitive
	match-value	
	new-value	

This is a sample of the result:

```

Request-Line: INVITE sip:servi ce@192.168.200.60:5060; tgi d=123 SIP/2.0
Message Header
Via: SIP/2.0/UDP
192.168.200.61:5060; branch=z9hG4bK9g855p30cos08acgs6o0.1
From: sip <sip:sip@192.168.1.60:5060>; tag=SD99ri 601-1
To: sut <sip:servi ce@192.168.1.61:5060>
Call-ID: SD99ri 601-6f5691f6461356f607b0737e4039caec-06a3gu0
CSeq: 1 INVITE
Contact: <sip:sip@192.168.200.61:5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sip <sip:user@192.168.1.60:5060; up=abc>; hp=123
Params: sip <sip:sip1@192.168.1.60:5060>
Params: sip <sip:sip2@192.168.1.60:5060>
Edit: disp <sip:user@192.168.1.60:5060; c1=sip1; c2=sip2>
```

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 l pu with the value stored from the previous Params header
 - If any of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value aup
 - If all of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value apu
 - If the first Params headers does not match the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter not with the value 123
 - If the first Params headers matches the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter yes with the value 456

This is a sample of the configuration:

si p-mani pul ati on		
name	speci al Char	
header-rul e		
name	searchParams	
header-name	Params	
acti on	store	
compari son-type	pattern-rul e	
match-val ue	. *si p: (. +)@. *	
msg-type	request	
new-val ue		
methods	I NVITE	
el ement-rul e		
name	userName	
parameter-name		
type	uri -user	
acti on	store	
match-val -type	any	
compari son-type	case-sensi tive	
match-val ue		
new-val ue		
el ement-rul e		
name	emptyUri Param	
parameter-name	abc	
type	uri -param	
acti on	store	
match-val -type	any	
compari son-type	pattern-rul e	
match-val ue		
new-val ue		
header-rul e		

name	addUserLast
header-name	Edit
action	manipulate
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	LastParamUser
parameter-name	Ipu
type	uri-param
action	add
match-val-type	any
comparison-type	case-sensitive
match-value	
new-value \$searchParams[^].\$userName. \$0	
element-rule	
name	anyParamUser
parameter-name	apu
type	uri-param
action	add
match-val-type	any
comparison-type	pattern-rule
match-value	\$searchParams[~]
new-value	aup
element-rule	
name	allParamUser
parameter-name	apu
type	header-param
action	add
match-val-type	any
comparison-type	pattern-rule
match-value	\$searchParams[*]
new-value	apu
element-rule	
name	notParamYes
parameter-name	not
type	uri-param
action	add
match-val-type	any
comparison-type	pattern-rule
match-value	
! \$searchParams. \$emptyUri Param	
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. \$emptyUri Param	
new-value	456

This is a sample of the result:

```
Request-Line: INVITE sip:servi ce@192. 168. 200. 60: 5060; tgi d=123 SIP/2. 0
Message Header
Via: SIP/2. 0/UDP
192. 168. 200. 61: 5060; branch=z9hG4bK681m9t30e0qh6akgj 2s1. 1
From: sip <sip:sip@192. 168. 1. 60: 5060>; tag=SDchrc601-1
To: sut <sip:servi ce@192. 168. 1. 61: 5060>
Call-ID: SDchrc601-fcf5660a56e2131fd27f12fcbd169fe8-06a3gu0
CSeq: 1 INVITE
Contact: <sip:sip@192. 168. 200. 61: 5060; transport=udp>
Goodbye: Remove Me
Custom: This is my custom header
Display: sip <sip:user@192. 168. 1. 60: 5060; up=abc>; hp=123
Params: sip <sip:sip1@192. 168. 1. 60: 5060>
Params: sip <sip:sip2@192. 168. 1. 60: 5060>
Edit: disp
<sip:user@192. 168. 1. 60: 5060; lpu=sip2; apu=aup; not=123>; apu=aup
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
```

Example 9: Status-Line Manipulation

This section shows an HMR configuration set up for status-line manipulation.

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the 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	manip
name	
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
new-value
methods
element-rule

name          is183Code
parameter-name
type          status-code
action        store
match-val-type
comparison-type
match-value
new-value

header-rule

name          change183
header-name   @status-line
action        manipulate
comparison-type
match-value
msg-type
new-value
methods
element-rule

name          make699
parameter-name
type          status-code
action        replace
match-val-type
comparison-type
match-value
new-value

sip-interface
options dropResponse=699

```

Example 10: Use of SIP HMR Sets

The following example shows the configuration for SIP HMR with one SIP manipulation configuration loading another SIP manipulation configuration. The goals of this configuration are to:

- Add a new header to an INVITE
- Store the user portion of the Request URI
- Remove all Route headers from the message only if the Request URI is from a specific user

```

sip-manipulation
name          deleteRoute
description   delete all Route Headers
header-rule

name          deleteRoute
header-name   Route
action        delete
comparison-type
match-value
msg-type
new-value
request

```

methods	INVITE
sip-manipulation	
name	addAndDelete
description	Add a New header and delete Route headers
header-rule	
name	addHeader
header-name	New
action	add
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	"Some Value"
methods	INVITE
header-rule	
name	storeURI
header-name	request-uri
action	store
comparison-type	pattern-rule
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	storeUser
parameter-name	
type	uri-user
action	store
match-val-type	any
comparison-type	pattern-rule
match-value	305.*
new-value	
header-rule	
name	deleteHeader
header-name	request-uri
action	sip-manip
comparison-type	Boolean
match-value	\$storeURI.\$storeUser
msg-type	request
new-value	deleteRoute
methods	INVITE

Example 11: Use of Remote and Local Port Information

The following example shows the configuration for remote and local port information. The goals of this configuration are to:

- Add LOCAL_PORT as a header parameter to the From header
- Add REMOTE_PORT as a header parameter to the From header

sip-manipulation	addorigin
name	addParam
description	
header-rule	
name	addParam
header-name	From
action	manipulate
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	addParam
parameter-name	newParam
type	header-param
action	add

match-val -type	any
comparison-type	case-sensitive
match-value	
new-value	\$LOCAL_IP
element-rule	
name	addLocalPort
parameter-name	lport
type	header-param
action	add
match-val -type	any
comparison-type	case-sensitive
match-value	
new-value	\$LOCAL_PORT
element-rule	
name	addRemotePort
parameter-name	rport
type	header-param
action	add
match-val -type	any
comparison-type	case-sensitive
match-value	
new-value	\$REMOTE_PORT

Example 12: Response/Status Processing

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the Net-Net SD 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	manip
name	
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      new-value

    header-rule
        name           change183
        header-name   @status-line
        action         manipulate
        comparison-type case-sensitive
        match-value
        msg-type       reply
        new-value
        methods
        element-rule
            name           make699
            parameter-name
            type           status-code
            action         replace
            match-val-type any
            comparison-type pattern-rule
            match-value    $IsContentLength0
                            & $is183.$is183Code
            new-value      699

    sip-interface
        options dropResponse=699

```

The following four configuration examples are based on the this sample SIP INVITE:

```

INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed; boundary=boundary
Content-Length: 466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
S=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap: 8 G729/8000/1
a=fmtp: 18 annexb=no
a=sendrecv
a=ptime: 20
a=maxptime: 200

```

```
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
a=ptime:30

--boundary--
```

Example 13: Remove a Line from SDP

In this example, the SIP manipulation is configured to remove all p-time attributes from the SDP.

sip-manipulation	removePtimeFromBody
name	removes ptime attribute from all bodies
description	
header-rule	
name	CTimeManip
header-name	Content-Type
action	manipulate
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	remPtime
parameter-name	application/sdp
type	mime
action	find-replace-all
match-val-type	any
comparison-type	case-sensitive
match-value	a=ptime:[0-9]{1,2}(\n \r\n)
new-value	

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060; branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>; tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060; user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed; boundary=boundary
Content-Length: 466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
```

```

t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap: 18 G729/8000/1
a=fmtp: 18 annexb=no
a=sendrecv
a=maxptime: 200

--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap: 34 H263a/90000

--boundary-

```

Example 14: Back Reference Syntax

In this sample of back-reference syntax use, the goal is to change the To user. The SIP manipulation would be configured like the following:

sip-manipulation	
name	changeToUser
description	change user in the To header
header-rule	
name	ChangeHeader
header-name	To
action	manipulate
comparison-type	case-sensitive
match-value	
msg-type	request
new-value	
methods	INVITE
element-rule	
name	replaceValue
parameter-name	
type	header-value
action	replace
match-value-type	any
comparison-type	pattern-rule
match-value	(.+) (service) (.+)
new-value	\$1+Bob+\$3

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```

INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060; branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>; tag=1
To: sut <sip:Bob@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060; user=phone>

```

```

Max-Forwards: 70
Content-Type: multipart/mixed; boundary=boundary
Content-Length: 466
...
...
...

```

Example 15: Change and Remove Lines from SDP

In this sample of changing and removing lines from the SDP, the goal is to convert the G.729 codec to G.729a. The SIP manipulation would be configured like the following:

<pre> sipp-manipulation name description proprietary-codec-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 </pre>	<pre> std2prop-codec-name rule-to-translate-standard-to CTypeMap Content-Type manipulate case-sensitive any g729-annexb-no-std2prop application/sdp mime find-replace-all any case-sensitive a=rtpmap: [0- G729a/8000/1 </pre>
---	--

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```

INVITE sipp: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 <sipp:sipp@192.168.1.60:5060>; tag=1
To: sut <sipp:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sipp:sipp@192.168.1.60:5060; user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed; boundary=boundary
Content-Length: 466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0

```

```

m=audio 12345 RTP/AVP 8
a=rtpmap: 18 G729a/8000/1
a=sendrecv
a=maxptime: 200

--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap: 34 H263a/90000

--boundary-

```

Example 16: Change and Add New Lines to the SDP

In this sample of changing and adding lines from the SDP, the goal is to convert non-standard codec H.263a to H.263. The SIP manipulation would be configured like the following:

sip-manipulation	name	prop2std-codec-name
	description	rule to translate proprietary
	to standard codec name	
	header-rule	
	name	CodecMap
	header-name	Content-Type
	action	manipulate
	comparison-type	case-sensitive
	match-value	
	msg-type	any
	new-value	
	methods	
	element-rule	
	name	H263a-prop2std
	parameter-name	application/sdp
	type	mime
	action	find-replace-all
	match-value-type	any
	comparison-type	case-sensitive
	match-value	a=rtpmap: ([0-
		9]{1,3}) H263a/. */\r\n
	new-value	a=rtpmap: +\$1+"
		H263/90000"+\$CRLF+a=fmtpt: +"\$1+" QCI F=4"+\$CRLF

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```

INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60

```

```

CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060; user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed; boundary=boundary
Content-Length: 466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 8
a=rtpmap: 18 G729/8000/1
a=fmtp: 18 annexb=no
a=sendrecv
a=maxptime: 200

--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap: 34 H263/90000
a=fmtp: 34 QCI F=4

--boundary-

```

Dialog-Matching Header Manipulation

The most common headers to manipulate using HMR are the To-URI and From-URI. Along with the to-tag, from-tag, and Call-ID values, these are also all headers that represent dialog-specific information that must match the UAC and UAS to be considered part of the same dialog. If these parameters are modified through HMR, the results can be that the UAC or UAS rejects messages.

While it is possible to ensure that dialog parameters match correctly using regular HMR, this feature offers a simpler and less error-prone method of doing so.

In addition, this section describes the addition of “built-in” SIP manipulations defined by 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 manipulating dialog-matching:

- Inbound HMR
- Outbound HMR

The new setting **out-of-dialog** (for the **msg-type** parameter) addresses these challenges by offering an intelligent more of dialog matching of messages for inbound and outbound HMR requests. This is a msg-type parameter, meaning that it becomes matching criteria for operations performed against a message. If you also specify methods (such as REGISTER) as matching criteria, then the rule is further limited to the designated method.

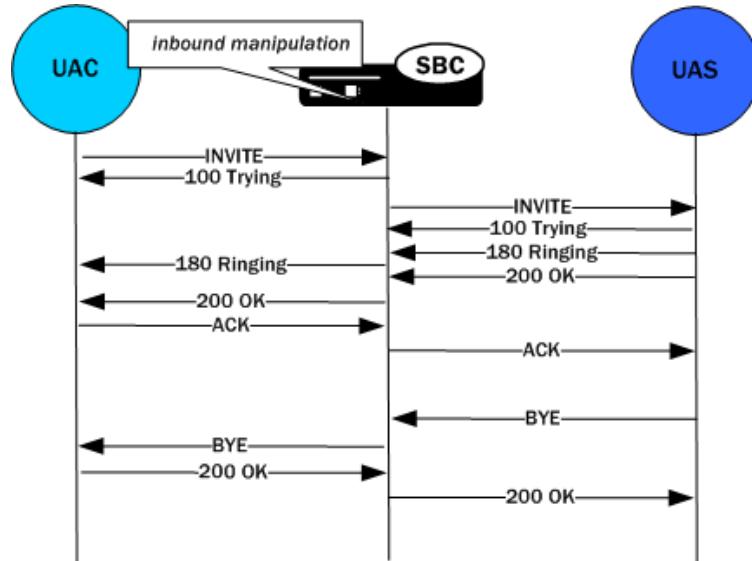
For both inbound and outbound manipulations, using the **out-of-dialog** setting means the message must be a request without a to-tag in order to perform the manipulation.

Inbound HMR Challenge

Since inbound manipulations take place before the message reaches the core of 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.

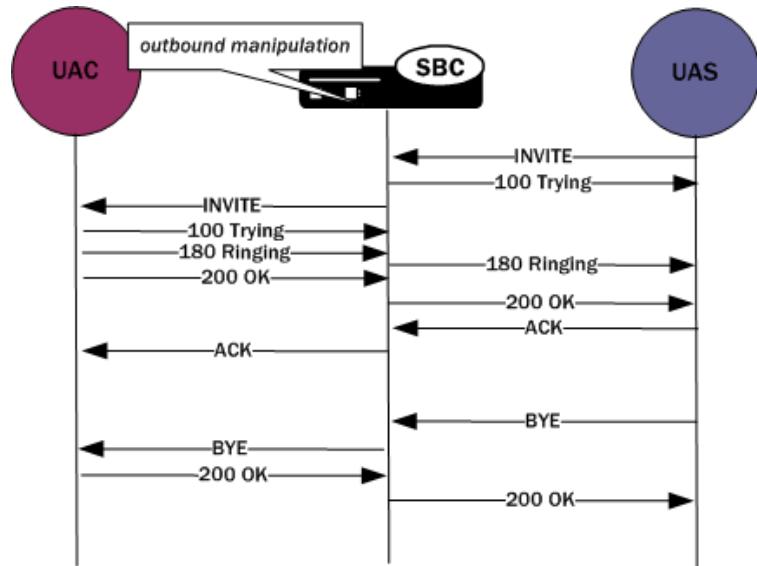
The Net-Net SBC performs the restoration of original headers outside of SIP manipulations. That is, there are no manipulation rules to configure for restore the header to their original context. The Net-Net SBC will recognize the headers have been modified, and restore them to their original state prior to sending the message out on the wire. Restoration takes place prior to outbound manipulations so that any outbound manipulation can those headers once they have been restored.



Outbound HMR Challenge

When you use the **out-of-dialog** setting for an outbound manipulation, the 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 use 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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 \(364\)](#) and [Outbound HMR Challenge \(364\)](#) sections for more information about how this setting works.

```
ACMEPACKET(sip-header-rules)# msg-type out-of-dialog
```
6. Save your work.

Built-In SIP Manipulations

In the course of HMR use, certain rules have become commonly used. Lengthy and complex, these rules do not include any customer-specific information and do they can be used widely. To make using them easier, they have been turned into built-in rules that you can reference in the **in-manipulationid** and **out-manipulationid** parameters that are part of the realm, session agent, and SIP interfaces configurations.

Built-in rules start with the prefix **ACME_**, so 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

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

```
3. Type **session-agent** and press <Enter>.

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

```
4. **out-manipulationid**—Enter name of the built-in rule you want to use. Remember that all built-in rules start with **ACME_**.
5. Save your work.

Testing SIP Manipulations

You can now use a new tool that allows you to test the outcome of your SIP manipulation and header rules without sending real traffic through the Net-Net SBC to see if they work.

To use the tool, you enter the ACLI's **test-sip-manipulation** utility and reference the rule you want to test using 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 **show msg 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 reserved words, such as: \$LOCAL_IP, \$TRUNK_GROUP_CONTEXT, and \$REMOTE_PORT. In addition, you can use your settings for testing across terminal sessions; if you choose to save your settings, everything (including the SIP message) will be saved, with the exception of the debugging option.

It is not recommended that you use this tool to add an ISUP message body.

HMR Import-Export

Due to the complexity of SIP manipulations rules and the deep understanding of system syntax they require, it is often difficult to configure reliable rules. This feature provides support for importing and exporting pieces of SIP manipulation configuration in a reliable way so that they can be reused.

Exporting

The SIP manipulation configuration contains an **export** command. When you use it, the Net-Net SBC sends the configuration you have selected to a designated file. The contents are the same information you see when you use the ACLI **show** command in XML format; it includes the selected configuration and any changes that have been made. Because you can only export one SIP manipulation configuration at a time, you must export each one-by-one if you need more than one.

The file name can be any you selected, and would be most useful if it were to identify its contents in some way. If the file already exists, then the export fails and informs you the file already exists. A successfully-executed export simply returns you to the system prompt.

The system writes exported files to `/code/imports`, a new location that will be created to avoid overlap with existing backup files. The files will carry the extension `.gz` to show that they have been compressed with gzip.

Your export data will look like this sample:

```
<?xml version='1.0' standalone='yes'?>
<simpManipulation
    name='manip'
    description=''
    lastModifiedBy='admin@console'
    lastModifiedDateTime='2009-10-16 14:16:29'>
    <headerRule
        headerName='Foo'
        msgType='any'
        name='headerRule'
        action='manipulate'
        cmpType='boolean'
        matchValue='$REGEX("[bB][A-Za-z]{2}")'
        newValue='foo'
        methods='INVERSE'>
    </headerRule>
</simpManipulation>
```

To avoid conflict with other objects on the system, key and object ID are not included as part of the exported XML.

Importing

Using the import command in the SIP manipulation configuration, you can import data from an exported file to a currently-selected configuration. If you have not selected a configuration into which to load the data, a new one will be created. Including the .gz extension, you enter the full name of the file you want imported. After it finds the file, the Net-Net SBC unarchives it and parses its contents. If these steps fail, the Net-Net SBC will alert you. If they succeed, then the configuration data loads into the object.

If you have been making changes to the configuration into which data was imported, the Net-Net SBC will inform you prior to importing the data so that you will not lose any of your work. This way, you will be less likely to overwrite unsaved changes.

Once the import is complete, it will be as if you entered the configuration by hand. You only need to save your work (by typing **done**) to save the SIP manipulation to the global SIP configuration. Note that if for some reason the XML is malformed or contained more than one object, the import will fail.

If you attempt to import a configuration with the same key as one that already exists, the system returns an error and prevents you from saving the imported object. In this case, you can delete the object with the same key and then carry out your import, or you can select the object with the same key and perform an import that will overwrite it with new data.

Displaying Imports

You can display imported SIP manipulations data at the system prompt. The command lists all files in the exported files directory, and also tells you if there are none.

Using FTP to Move Files

You can also place exported SIP manipulation configuration files on the Net-Net SBC using FTP. You need to use the same /code/imports directory to do so.

Removing Files

Using the **delete-import** command with the name of the file you want to delete removes it from the system. Using this command, you can delete files that are no longer useful to you. Carrying out this command is final and there is no warning before you go ahead with the deletion. A failed deletion (for instance, because there is no such file) will produce an error message; a successful deletion simply returns you to the system prompt.

Unique HMR Regex Patterns and Other Changes

In addition to the HMR support it offers, the 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 `$MANI P_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 `$MANI P_PATTERN` is used in a manipulation rule's **new-value** parameter, it resolves to an empty string, equivalent of no value. Even though this process ends with no value, it still consumes system resources. And so Acme Packet recommends you do not use `$MANI P_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	publ i c: 0	
...		
manipulation-pattern	Lorem(.+)	
sip-manipulation		
name	manip	
description		
header-rules		
name	headerRule	
header-name	Subj ect	
action	mani pul ate	
match-value	\$MANI P_PATTERN	
msg-type	request	
comparison-type	pattern-rule	
new-value	Math	
methods	INVI TE	

Reject Action

Release S-C6.2.0 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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 sipd** display, refer to the Rejected Messages category; there is no distinction between requests and responses.

```
ACMEPACKET# show sipd
13:59:07-102
SIP Status          -- Period -- ----- Lifetime -----
                    Active   High    Total    Total  PerMax  High
Sessions           0        0       0        0       0       0
Subscriptions      0        0       0        0       0       0
Dialogs            0        0       0        0       0       0
CallID Map         0        0       0        0       0       0
Rejections          -        -       0        0       0       0
```

Rel NVI TEs	-	-	0	0	0	0
Media Sessions	0	0	0	0	0	0
Media Pending	0	0	0	0	0	0
Client Trans	0	0	0	0	0	0
Server Trans	0	0	0	0	0	0
Resp Contexts	0	0	0	0	0	0
Saved Contexts	0	0	0	0	0	0
Sockets	0	0	0	0	0	0
Req Dropped	-	-	0	0	0	0
DNS Trans	0	0	0	0	0	0
DNS Sockets	0	0	0	0	0	0
DNS Results	0	0	0	0	0	0
Rejected Msgs	0	0	0	0	0	0

Session Rate = 0.0
 Load Rate = 0.0
 Remaining Connections = 20000 (max 20000)

SNMP Support

The 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."
        ::= { apSysMgmtMI BGeneral Obj ects 18 }

apSysMgmtRejectedMessagesThresholdExceededTrap      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 {
        apSysMgmtRejectedMessagesThresholdExceededTrap
    }
    STATUS         current
    DESCRIPTION
        "Traps used for notification of rejected messages"

```

```

 ::= { apSystemManagementNotificationsGroups 26 }

apSmgmtRejectedMessagesCap
AGENT-CAPABILITYIES
PRODUCT-RELEASE      "Acme Packet SD"
STATUS                current
DESCRIPTION          "Acme Packet Agent Capability for
                     enterprise system management MIB."
SUPPORTS             APSYSGMT-MIB
INCLUDES              {
                     apSysMgmtRejectedMessagesGroup,
                     apSysMgmtRejectedMessagesNotificationsGroup
}
 ::= { apSmgmtMibCapabilities 37 }

```

Log Action

Release S-C6.2.0 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[checkRURI Port]
INVITE sip:servi ce@192.168.1.84:5060 SIP/2.0
From: sip <sip:+2125551212@192.168.1.60:5060>;tag=3035SIPpTag001
To: sut <sip:servi ce@192.168.1.84>
Contact: sip:sipp@192.168.1.60:5060
-----
```

Changes to Storing Pattern Rule Values

Release S-C6.2.0 introduces changes to the framework for storing regular expression results within manipulation rules, altering the way the **store** action works. These changes are beneficial to performance.

In previous releases, when the **store** action is used, the 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 S-C6.2.0:

- The action **find-replace-all** now executes all element rules. Previously, no child rules were executed.
- The action **sip-manip** now executes existing all element rules. Previously, no child rules were executed.
- The action **store** now executes existing all element rules. Previously, only child rules with the **store** action were executed.
- The action **add** now executes existing all element rules. Previously, only child rules with the **add** action were executed.

Name Restrictions for Manipulation Rules

Historically, you have been allowed to configure any value for the name parameter within a manipulation rule. This method of naming caused confusion when referencing rules, so now manipulation rules name must follow a specific syntax. They must match the expression "`^[[alpha:]][[:alnum:]_]+$`" and contain at least one lower case letter.

In other words, the name must:

- Start with a letter, and then it can contain any number of letters, numbers, or underscores
- Contain at least one lower case letter

All pre-existing configurations will continue to function normally. If you want to change a manipulation rule, however, you are required to change its name if it does not follow the new format.

The ACLI **verify-config** command warns you if the system has loaded a configuration containing illegal naming syntax.

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 used will have no impact to normal SIP manipulation operations.

Dialog Transparency

This section explains how to configure dialog transparency, which prevents the Net-Net SBC from generating a unique Call-ID and modifying dialog tags.

Overview

With dialog transparency enabled, the Net-Net SBC is prevented from generating a unique Call-ID and from modifying the dialog tags; the Net-Net SBC passes what it receives. Therefore, when a call made on one Net-Net SBC is transferred to another UA and crosses a second Net-Net SBC, the second Net-Net SBC 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 SBC or how many Net-Net SBCs a call crosses.

Without dialog transparency enabled, the Net-Net SBC's SIP B2BUA rewrites the Call-ID header and inserts 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 SBC can recognize the direction of a dialog. However,

this behavior makes call transfers problematic because one Net-Net SBCs' Call-ID might not be properly decoded by another Net-Net SBC. The result is asymmetric header manipulation and failed call transfers.

Configuring Dialog Transparency

ACLI Instructions and Examples

You set one parameter in your SIP configuration to enable dialog transparency.

- For new configurations, this feature defaults to enabled

To enable SIP dialog transparency:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the **session-router** path.
ACMEPACKET(configure)# **session-router**
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-config**
4. Use the ACLI **select** command so that you can work with the SIP configuration.
ACMEPACKET(sip-config)# **select**
5. **dialog-transparency**—Enter the state of SIP dialog transparency you require for your Net-Net SBC. The default value is **enabled**. The valid values are:
 - enabled | disabled

Route Header Removal

This section explains how to enable the Net-Net SBC to disregard and strip all SIP Route headers. You set an option in a SIP interface configuration to strip all Route headers for SIP requests coming from this interface.

When the Net-Net SBC with this option configured receives an INVITE from an interface, it removes the route headers. However, although it removes the headers, the Net-Net SBC maintains backward compatibility with RFC 2543 nodes. To do so, it normalizes the request to an RFC 3261 loose routing form before it removes the headers.

Configuring SIP Route Header Removal

ACLI Instructions and Examples

The following information explains how to remove SIP route headers.

To configure SIP route header removal:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-interface**
ACMEPACKET(sip-interface)#
4. Type **options strip-route-headers** and press <Enter>. This completes the configuration of SIP route header removal.
ACMEPACKET(sip-interface)# **options strip-route-headers**

SIP Via Transparency

This section explains the inbound Via header transparency feature, which enables the Net-Net SBC 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 SBC still replaces the Contact and other header addresses with its own, and does not pass on the core's Via headers in outbound requests.

This feature is targeted for the Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) with SIP hosted NAT traversal support. It works with SIP NAT bridged, local-policy routed, and non-SIP NAT configurations, regardless of registration handling.

How it Works

Some equipment acts as Proxy-CSCF (P-CSCF) and Serving-CSCF (S-CSCF) nodes, with the Net-Net SBC is located between the equipment and user endpoints. The equipment needs to see the each user endpoint's original Via header in order to perform some implicit authentication, admission, and control functions in a TISPAN-compliant model.

You enable Via header transparency on the access SIP interface. Received Via headers are saved for inclusion in requests going out another interface or session agent that does not have the parameter set, in other words, the core side. For any received SIP message where the inbound previous hop interface was enabled for Via header transparency, the Net-Net SBC 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 SBC also adds a received= parameter to the copied Via header, per the SIP RFC 3261.

Any message received from an interface without Via header transparency enabled, does not have the received Via header copied over to any other direction.

For HNT, where the original top-most (and only) Via header from a UE is a private/false address, the SD should still copy that false address into the core-side, and the received= parameter will contain the real Layer-3 addressing.

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)#[/list]

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 your work using the ACLI **save** or **done** command.

Symmetric Latching

Symmetric latching, or forced HNT, ensures that symmetric RTP/RTCP is used for a SIP endpoint. Symmetric RTP/RTCP means that the IP address and port pair used by an outbound RTP/RTCP flow is reused for the inbound flow. The IP address and port are learned when the initial RTP/RTCP flow is received by the Net-Net SBC. The flow's source address and port are latched onto and used as the destination for the RTP/RTCP sourced by the other side of the call. The IP address and port in the c line and m line respectively in the SDP message are ignored.

If your network is configured with nested realms in order to separate signalling from media, make sure that the symmetric latching feature is enabled on the signaling realm.

Note: This description is applicable to RTCP only when you also enable the HNT RTCP option in the **media-manager** configuration. Do not enable symmetric latching on core-facing interfaces.

ACLI Instructions and Examples

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(config)# 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(real-m-config)#

```
4. Select the realm where you want to apply this feature.


```
ACMEPACKET(real-m-config)# select
Identifier:
1: Acme_Realm <none>           0.0.0.0
```

2: MGCP_Real m <none> 0. 0. 0. 0
3: H323REALM <none> 0. 0. 0. 0

selection: 1
ACMEPACKET(real m-confi g) #

5. **symmetric-latching**—Enable symmetric latching on the Net-Net SBC. This completes the configuration of forced HNT. The default value for this parameter is **disabled**. The valid values are:

- enabled | disabled

ACMEPACKET(real m-confi g) # **symmetric-latching enabled**

6. Save your work using the ACLI **save** or **done** command.

Enabling RTCP Latching

To enable RTCP symmetric latching:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
 2. Type **media-manager** and press <Enter> to access the media-level configuration elements.
ACMEPACKET(configure)# **media-manager**
ACMEPACKET(media-manager)#
 3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# **media-manager**
ACMEPACKET(media-manager-config)#
 4. Select the media manager configuration so that you can enable HNT RTCP.
ACMEPACKET(media-manager-config)# **select**
 5. **hnt-rtcp**—Enable support of RTCP when the Net-Net SBC performs HNT. The default value is **disabled**. The valid values are:
 - enabled | disabledACMEPACKET(media-manager-config)# **hnt-rtcp enabled**
 6. Save your work using either the ACLI **save** or **done** command.

SIP Number Normalization

This section explains the SIP number normalization feature that applies to the SIP To URI. (Currently the Net-Net SBC supports number normalization on From and To addresses for both inbound and outbound call legs.) Number normalization includes add, delete, and replace string functions that result in consistent number formats.

Number normalization is supported for the following call types:

- SIP to SIP
 - H.323 to SIP

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 SBC uses the following number normalization rules:

- North American Numbering Plan (NANP) calls: where a number with the format 1NPANXXXXXX is received, the Net-Net SBC adds a plus sign (+) as a prefix to the NANP number. The Net-Net SBC also adds the string ; user=phone after the host IP address in the SIP URI. For example:

```
si p: +1NPANXXXXXX@i paddr; user=phone
```
- International NWZ1 calls: Net-Net SBC receives an international call with the format 011CCNN. The Net-Net SBC 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:

```
si p: +CCNN@i paddr; 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 SBC.
- Calls to numbers such as N11, 0-, 0+, 00-, and 01+: the Net-Net SBC 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:

```
si p: N11; phone-context=+1@i paddr; user=phone
si p: 01CCNN; phone-context=+1@i paddr; user=phone
```
- Calls with numbers that are already normalized are not modified by the Net-Net SBC.

Calls from IP Peer Network

For calls received from external peer networks, the Net-Net SBC 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 SBC 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 SBC 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

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(real-m-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(real-m-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(real-m-config)# **select**
ACMEPACKET(real-m-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(real-m-config)# **select**
ACMEPACKET(real-m-config)# **options +number-normalization**
 5. Save your work using the ACLI **save** or **done** command.

Session Agent

To configure SIP number normalization for a session agent:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the media-level configuration elements.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#
 3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **session-agent**

```
ACMEPACKET(sessi on-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 session agent configuration, add the option by typing **options**, a <Space>, and then **number-normalization**.

```
ACMEPACKET(sessi on-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(sessi on-agent)# select
```

```
ACMEPACKET(sessi on-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(sessi on-agent)# select
```

```
ACMEPACKET(sessi on-agent)# options +number-normalization
```

5. Save your work using the ACLI **save** or **done** command.

SIP Port Mapping

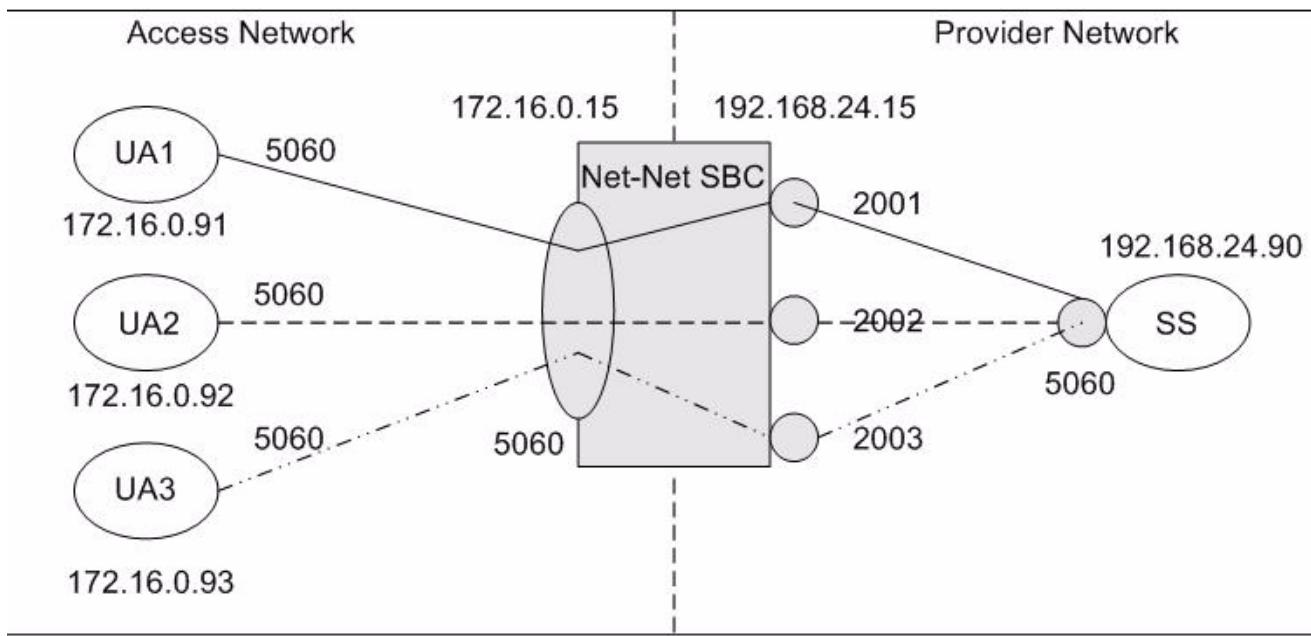
This section contains information about the SIP port mapping feature. SIP port mapping lets you allocate a unique SIP signaling transport address (IP address and UDP port) on the Net-Net SBC in the provider network for each registered endpoint (user agent).

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 SBC resides between the endpoints and the softswitch, the softswitch sees the same transport address (that of the Net-Net SBC) for all endpoints. By allocating a unique UDP port for each endpoint, the Net-Net SBC provides each of them a unique transport address.

The following example illustrates the SIP port mapping feature.



The diagram shows UA1, UA2, and UA3 are endpoints within the access network and that the SIP interface for the access network is 172.16.0.15:5060. On the provider network, the SIP interface is at 192.168.24.15, with the SIP port mapping feature enabled. The softswitch/registrar is also located on the provider network at 192.168.24.90:5060.

The diagram shows that port 2001 on the provider network is allocated to UA1 on the access network, port 2002 is allocated to UA2, and port 2003 is allocated to UA3. Because of this allocation, all SIP signaling messages sent from the endpoints in the access network to the softswitch on the provider network travel through an allocated signaling port. For example, all signaling messages between UA1 and the softswitch use 192.168.24.15:2001 as the transport address.

How SIP Port Mapping Works

The Net-Net SBC allocates SIP port mapping (signaling) ports during a REGISTER request that has registration caching applied. When you define a range of signaling ports for the SIP interface, you create a pool of signaling ports that can be allocated during the REGISTER request.

The Net-Net SBC allocates a signaling port from the pool when it creates the registration cache entry for a Contact in a REGISTER request. It allocates a separate signaling port for each unique Contact URI from the access side. The registration cache Contact entry contains the mapping between the Contact URI in the access/endpoint realm (the UA-Contact) and the Contact URI in the registrar/softswitch realm (the SD-Contact).

The SD-Contact is the allocated signaling port. The signaling port gets returned to the pool when the Contact is removed from the registration cache. The removal can occur when the cache entry expires; or when the endpoint sends a REGISTER request to explicitly remove the Contact from the registrar. When a signaling port returns to the pool it gets placed at the end of pool list; in a least-recently-used allocation method for signaling ports

When the Net-Net SBC forwards the REGISTER request to the softswitch, it replaces the UA-Contact with SD-Contact. For example, if UA1 sends a REGISTER request with a Contact URI of `si p: ua1@172. 16. 0. 91: 5060`, it is replaced with `si p: 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 SBC still processes requests received on the configured SIP port address. Requests sent into the registrar/softswitch realm that are not associated with a registered user will use the configured SIP port address.

Using SIP port mapping with SIPconnect—where unique ports are used for each registered PBX—hinders the Net-Net SBC from routing incoming calls to the corresponding PBX because the Net-Net SBC uses DN for the PBX's parent during registration, but the incoming INVITE from the softswitch contains the child DN in its Request URI. Thus the Net-Net SBC cannot find a matching SBC-Contact because the username of the Request URI contains the child DN, but the username of the SBC-Contact contains the parent DN.

You can enable SIPconnect support in either the realm configuration or session agent for the SIP access network by setting the `sip-connect-pbx-reg` option. With this option set and the destination realm configured for port mapping, the Net-Net SBC inserts a special search key in the registration table. Rather than adding the SD-Contact as the key as with regular (non-SIPconnect) registrations, the Net-Net SBC strips user information and instead uses the host and port information as the registration key. The Net-Net SBC still forwards the registration message with an intact contact username.

SIP Port Mapping Based on IP Address

Some registrars need to know that multiple contacts represent the same endpoint. The extension to this feature answers the expectation from registrars that an endpoint registering multiple AoRs will use a single core-side mapped port to show that the AoRs really represent a single endpoint.

When you enable SIP port mapping based on IP Address, the Net-Net SBC supports core-side UDP port mapping based on the endpoint's IP address. It ignores the username portion of the AoR or Contact.

The Net-Net SBC performs the port mapping allocation and lookup based on all requests using the via-key from the SIP Request. The via-key is a combination of Layer 3 and Layer 5 IP information in the message. The Net-Net SBC performs an additional lookup in the registration table to determine if a via-key already exists. If it does, then the Net-Net SBC uses the port already allocated and does not allocate a new one.

About NAT Table ACL Entries

To enable SIP signaling messages to reach the host processor, the Net-Net SBC 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:

i	ntf:	vlan	source-ip/mask:	port/mask	dest-ip/mask:	port/mask	prot	type	i	ndex
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:

i	ntf:	vlan	source-ip/mask:	port/mask	dest-ip/mask:	port/mask	prot	type	i	ndex
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.

Registration Statistics

The SIP registration cache statistics include counters for free and allocated signaling ports. You can issue a show registration command to display the statistics:

SIP Registrations		-- Period --		Lifetime -----		
	Active	High	Total	Total	PerMax	High
User Entries	4	4	0	7	4	4
Local Contacts	4	4	0	7	4	4
Free Map Ports	12284	12284	0	12291	12288	12288
Used Map Ports	4	4	0	7	4	4
Forwards	-	-	1	22	4	4
Refreshes	-	-	3	43	3	3
Rejects	-	-	0	0	0	0
Timeouts	-	-	0	1	1	1
Fwd Postponed	-	-	0	0	0	0
Fwd Rejected	-	-	0	0	0	0
Refr Extension	0	0	0	0	0	0
Refresh Extended	-	-	0	0	0	0

The labels for the first two items reflect the restructured registration cache:

- User Entries: counts the number of unique SIP addresses of record in the cache. Each unique address of record represents a SIP user (or subscriber). The address of record is taken from the To header in the REGISTER request. There might be one or more registered contacts for each SIP user. The contacts come from the Contact header of the REGISTER request.
- Local Contacts: counts the number of contact entries in the cache. Because the same user can register from multiple endpoints (user agents); the number of Local Contacts might be higher than the number of User Entries.
- Free Map Ports: counts the number of ports available in the free signaling port pool.
- Used Map Ports: counts the number of signaling ports allocated for registration cache entries. The value of Used Map Ports will equal the number of Local Contacts when the port mapping feature is used for all registrar/softswitch realms in the Net-Net SBC.

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)# sessi on-router
```

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

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

```

4. **port-map-start**—Set the starting port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. The default values is 0 and when this value is set, SIP port mapping is disabled. The valid range is:
 - Minimum: 0, 1025
 - Maximum: 65535

```
ACMEPACKET(sip-interface)# port-map-start 32768
```

5. **port-map-end**—Set the ending port for the range of SIP ports available for SIP port mapping. The valid range is 1025 through 65535. If you set the value to the default 0, SIP port mapping is disabled. The valid range is:
 - Minimum—0, 1025
 - Maximum—65535

Note: If not set to zero (0), the ending port must be greater than the starting port.

```
ACMEPACKET(sip-interface)# port-map-end 40959
```

6. **options**—If you want to use SIP port mapping based on IP address, set the options parameter by typing **options**, a <Space>, the option name **reg-via-key** with a “plus” sign in front of it, type the equal sign and the word **all**. Then press <Enter>.

```
ACMEPACKET(sip-interface)# options +reg-via-key=all
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

7. Save your work using the ACLI **done** command.

The following example shows SIP port mapping configured for a SIP interface:

sip-interface	
state	enabl ed
real-m-id	backbone
sip-port	
address	192.168.24.15
port	5060
transport-protocol	UDP
allow-anonymous	all
sip-port	
address	192.168.24.15
port	5060
transport-protocol	TCP
allow-anonymous	all
carriers	
proxy-mode	
redirection-action	
contact-mode	
nat-traversal	none

nat-interval	30
registration-caching	enabled
min-reg-expire	120
registration-interval	3600
route-to-registry	enabled
tel-uri-scheme	disabled
uri-fqdn-domain	
trust-mode	agents-only
max-nat-interval	3600
nat-int-increment	10
nat-test-increment	30
sip-dynamic-hnt	disabled
stop-recurse	401, 407
port-map-start	32768
port-map-end	40959
Last-modified-date	2005-09-23 14:32:15

SIP Port Mapping for TCP and TLS

In releases prior to S-C6.2.0, the Net-Net SBC supports SIP port mapping for UDP and now you can enable this feature for SIP sessions using TCP and TLS. Port mapping enables the Net-Net SBC to allocate a unique port number for each endpoint registering through it by giving it a transport address (or hostport) in the registered Contact.

When you enable this feature for TCP and TLS, the Net-Net SBC designates a port from a configured range for each endpoint that registers with SIP servers in the SIP interface's realm. You establish that range of ports using the **port-map-start** and **port-map-end** parameters. Unlike its behavior with UDP port mapping—where the Net-Net SBC sends requests on the SIP interface from the allocated port mapping, the Net-Net SBC sends all requests over an existing connection to the target next hop for TCP/TLS port mapping. If a connection does not exist, the system creates one. So for TCP/TLS port mapping, only the Contact header contains the transport address of the mapping port (i.e., the transport address of the configured SIP port). And the system refuses TCP and TLS connections on the allocated mapping port.

With TCP/TLS port mapping enabled, the Net-Net SBC sends the Path header with the transport address in Register requests, unless you specify that it should not do so. Standards-conformant SIP servers (that support RFC 3327) might attempt to send requests to the allocated mapping port if the Path header is absent.

Note: ACL entries in the NAT table that permit TCP/TLS signaling for a SIP port configuration with TCP/TLS port mapping are the same as they would be for a TCP/TLS SIP port without port mapping enabled. Additional ACL entries that need to be set up for UDP port mapping are not required for TCP/TLS port mapping.

RTN 1684

ACLI Instructions and Examples

You enable TCP/TLS port mapping in a per-realm basis using the SIP interface configuration; setting the **tcp-port-mapping** value in the **options** parameter enables the feature. Enabling this parameter turns on the port mapping feature for UDP as well.

By default, the Net-Net SBC includes the Path header in Register requests it sends from that SIP interface. If you do not want this header to be included, however, you can set the value as **tcp-port-mapping=nopath**.

To enable TCP/TLS port mapping for a SIP interface:

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

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

```

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

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

```

3. Type **sip-interface** and press <Enter>. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.

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

```

4. **options**—Set the options parameter by typing options, a <Space>, the option name **tcp-port-mapping** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(sip-interface)# options +tcp-port-mapping
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save your work.

SIP Configurable Route Recursion

When the Net-Net SBC 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 SBC 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 SBC, the 3xx response can include multiple Contacts to which the request should be forwarded in a specific order. In both cases, the Net-Net SBC needs to recurse through a list of targets.

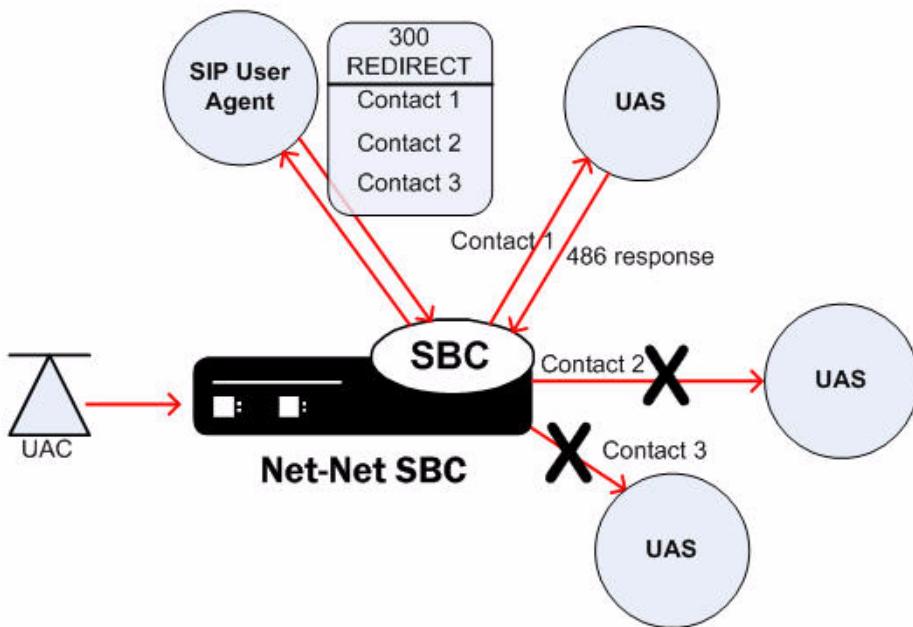
When the Net-Net SBC receives a non-successful (or non-6xx response) final response from the UAS, and there are multiple targets for the original request, the SD will forward the request to the next target and wait for a response. While the process of forwarding the request to multiple targets as explained 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 SBC 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 SBC 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 SBC 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 SBC 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

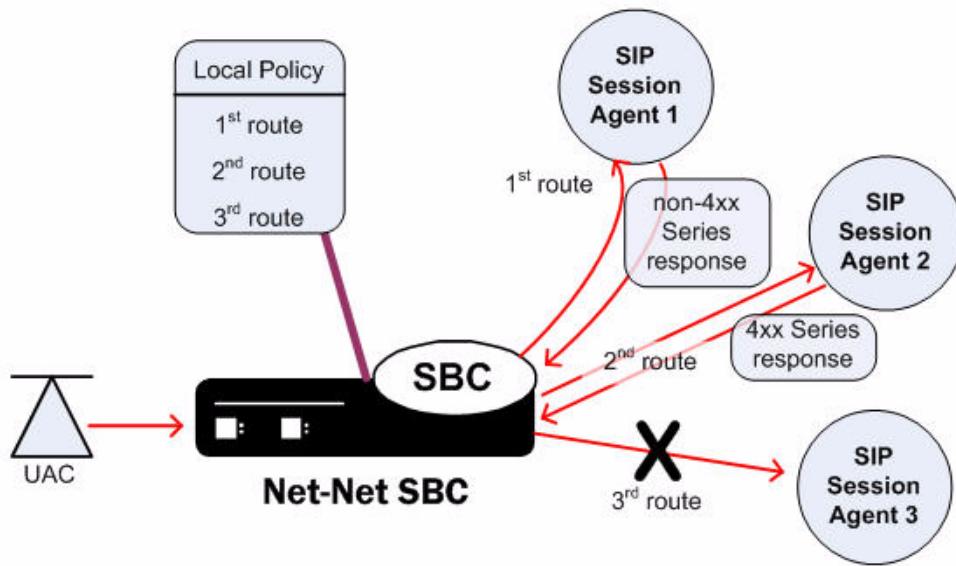
A more detailed example is when a softswitch might return a list of contacts for multiple PSTN gateways in a Redirect message. If the PSTN target number contacted on redirection is busy, a 486 response will be sent to the Net-Net SBC. Since the single target is located in the PSTN, a subsequent request through a different gateway will yield another 486 response. The Net-Net SBC 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 SBC might determine from a local policy lookup that several routes are applicable for forwarding a SIP message. The Net-Net SBC 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 SBC. 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

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# **config 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 real m= ip=1.0.0.0

```
2: SI PSA      real m=      ip=10.10.102.1
```

```
selection: 2
```

```
ACMEPACKET(session-agent) #
```

5. **stop-recuse**—Enter list of returned response codes that this session agent will watch for in order to stop recursion on the target's or contact's messages. This can be a comma-separated list or response code ranges.
ACMEPACKET(session-agent) # **stop-recuse 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**
<real m-id>
1: Acme_Real m
selection: 1
ACMEPACKET(sip-interface) #
5. **stop-recuse**—Enter a list of returned response codes that this SIP interface will watch for in order to stop recursion on the target's or contact's messages. This list can be a comma-separated list of response codes or response code ranges.
ACMEPACKET(sip-interface) # **stop-recuse 404, 484-486**
6. Save and activate your changes.

SIP Event Package Interoperability

Service providers often deploy a Net-Net SBC 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 SBC 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 SBC 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.ietf.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](http://www.ietf.org/info/players/ietf/presence/outdated/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# config terminal
2. Type **session-router** and press <Enter>
ACMEPACKET(config)# 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 SBC does not forward the REGISTER to the registrar, because it believes the endpoint is already registered, based on a previous registration from the same Contact: header URI.

To remedy this problem, the Net-Net SBC now keeps track of the Call-ID in its registration cache. The **forward-reg-callid-change** option in the global SIP configuration element forces the Net-Net SBC to forward a REGISTER message to the registrar when the Call-ID header changes in a REGISTER message received from a reregistering UAC.

ACLI 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 has been added as an enhancement to the SIP response code mapping. The SIP response code map feature lets you establish a table that maps SIP response-received messages (entries) to response-to-send messages (entries).

SIP local response code mapping is used with the SIP responses generated by the Net-Net SBC towards a specific SIP session agent. This feature lets you provision the

mapping of the response codes used by the Net-Net SBC when it generates the responses towards a session agent.

You create the SIP local response code map using the existing mapping functionality, and then assigning that map to a session agent or to a SIP interface.

Note: The configured response map is not used when the Net-Net SBC is acting as proxy for the responses to this session agent.

ACLI Instructions and Examples

Creating a SIP Response Code Map

The following instructions explain how to create the SIP response code map and then how to assign it to a specific session agent.

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
xmit-code
reason
local-error
```

5a. **recv-code**—Enter original SIP response code for the **recv-mode** parameter.

The valid range is:

- Minimum—100
- Maximum—699

```
ACMEPACKET(response-map-entries)# recv-mode 486
```

5b. **xmit-code**—Enter the SIP response code into which you want the original response code to be translated. This valid range is:

- Minimum—100
- Maximum—699

```
ACMEPACKET(response-map-entries)# xmit-mode 600
```

- 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.

```
ACMEPACKET(response-map-entries)# reason "Busy Everywhere"
```

- 5d. **local-error**—Enter the local error that triggers the use of this local response map. Supported values are:

- enum-void-route
- monthly-minutes-exceed
- next-hop-sa-oos
- recv-sa-exc-constraints
- recv-sip-int-exc-constraints
- next-hop-sa-exc-constraints
- next-hop-sip-int-exc-constraints
- realm-bw-exc-poly-serv-reject
- no-steering-pool-ports-available
- allow-anonymous-rejection

```
ACMEPACKET(response-map-entries)# local -error next-hop-sa-oos
```

6. Note the name that you gave the SIP response code map so that you can use it when you configure a session agent to support SIP response code mapping.
7. Save and activate your changes.

Assigning SIP Response Code Maps to Session Agents

To assign a SIP local response code map to a session agent:

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

```
ACMEPACKET# configure terminal
```

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

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

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

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

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

```
ACMEPACKET(session-agent)#[
```

4. **local-response-map**—Enter the name of the configured SIP response map that you want to use for this session-agent and press <Enter>.

```
ACMEPACKET(session-agent)# local -response-map busy
```

5. Save and activate your configuration.

Assigning SIP Response Code Maps to SIP Interfaces

To apply SIP response codes maps to a SIP interface:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

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

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

3. Type **sip-interface** and press <Enter>.
 ACMEPACKET(session-router)# **sip-interface**
 ACMEPACKET(sip-interface)#
4. **local-response-map**—Enter the name of the configured SIP response map that you want to apply to this SIP interface for locally-generated SIP responses. This parameter is blank by default.
5. Save and activate your configuration.

Session Agent Ping Message Formatting

You can configure the user portion of the From: header, the To: header, and the Request-URI in the ping-type message that the Net-Net SBC 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.

```
OPTI ONS si p: bob@si p. com SIP/2. 0
From: UA1 <si p: bob@si p. com>
To: NUT <si p: anna@gw. si p. com>
Call -ID: 123abc@desk. si p. com
CSeq: 1 OPTI ONS
Contact: <si p: UA1@cl i ent. si p. com>
Accept: appl i cati on/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.

ACLI Instructions and Examples

To configure session agent ping message formatting:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **confi gure termi nal**
2. Type **session-router** and press <Enter> to access the session-router configuration elements.
 ACMEPACKET(config)# **sessi on-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)# **sessi on-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**—Set the user portions of the Request-URI and the From: header in a session agent ping message.
 ACMEPACKET(session-agent)# **pi ng-from-user-part bob**

5. **ping-to-user-part**—Set the user portion for the To: header in a session agent ping message.
ACMEPACKET(session-agent)# **ping-to-user-part anna**
6. Save and activate your configuration.

SIP PAI Stripping

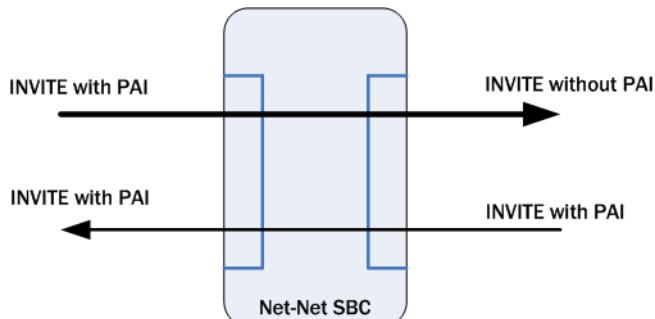
The Net-Net SBC now has the ability to strip P-Asserted-Identity (PAI) headers so that service providers can ensure an extra measure of security against malicious users pretending to be legitimate users. To pretend to represent another account, the malicious users simply send an INVITE with an imitation PAI. This feature allows real-time detection of such fraudulent use.

This feature uses a combination of:

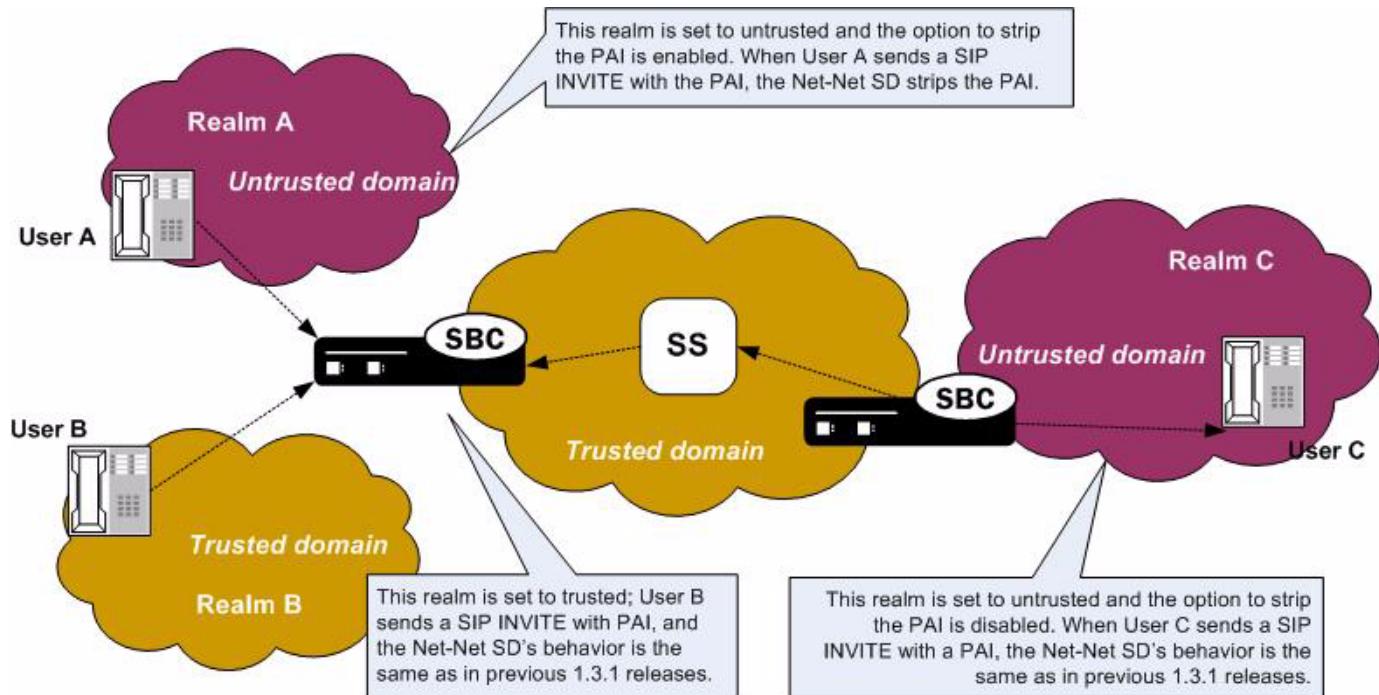
- DoS protection applied on a per-realm basis
- SIP PAI header stripping

The combination of these settings can produce different results for the SIP PAI stripping feature.

- *SIP PAI header stripping enabled for an untrusted realm*—If the PAI stripping parameter is set to enabled in a realm that is untrusted, then the Net-Net SBC strips the PAI headers from SIP INVITEs that are received from the external address, regardless of the privacy type. The Net-Net SBC then sends the modified INVITE (without the PAI). If the INVITE comes from a trusted realm, then the Net-Net SBC does not strip the PAI header and the Net-Net SBC behaves as it does when you are using previous 1.3.1 releases.



- *Multiple SIP PAIs in a SIP INVITE*—The Net-Net SBC removes all PAIs when there are multiple PAIs set in SIP INVITEs that come from untrusted realms.
- *Net-Net SBC behavior bridging trusted and untrusted realms*—The following graphics shows you how Net-Net SBCs can be positioned and configured to handle PAI stripping between trusted and untrusted realms.



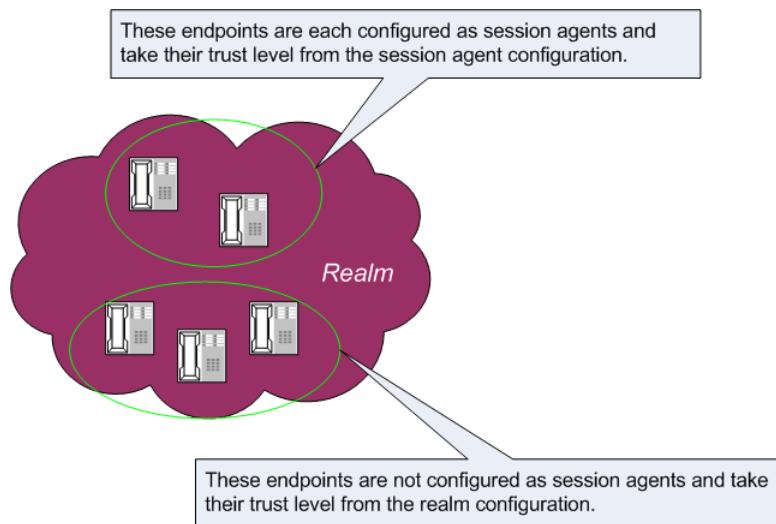
Realm Configuration Settings	REALM A	REALM B	REALM C
Realm designation trusted or untrusted (trust-me)	Disabled	Enabled	Enabled
SIP PAI stripping (pai-strip)	Enabled	Enabled or disabled	Disabled
Net-Net SBC's behavior	Strip PAI regardless of privacy type	Same as behavior for SIP privacy support in previous 1.3.1 releases	Same as behavior for SIP privacy support in previous 1.3.1 releases

SIP PAI Stripping Configuration

When you configure this feature, please note how the Net-Net SBC behaves when you combine the designation of a realm as trusted/untrusted and SIP PAI stripping is enabled. Enter the choices for the ACLI **trust-me** and **pai-strip** parameters accordingly.

Be aware that trust is also established in the session agent configuration, and that the trust level set in a session agent configuration overrides the trust set in a realm configuration. For example, a realm might have several endpoints, some of which are associated with session agents and some of which are not. The endpoints that have configured session agent will take their trust level from the session agent.

parameters you set; the other endpoints, ones that are not associated with session agents, take their trust level from the realm parameters you set.



Take this relationship into consideration when you configure SIP PAI header stripping, or this feature will not work as designed.

For the sample configuration cited below, the desired Net-Net SBC behavior is to always strip the PAI regardless of privacy type.

ACLI Instructions and Examples

To configure SIP PAI stripping for an existing realm using the ACLI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter> to access the media-manager path.
ACMEPACKET(configure)# **media-manager**
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# **realm-config**
ACMEPACKET(realm-config)#


```

      i denti fier:
      1: acmePacket <none>          192.168.20.0/24
      2: realm1      <none>          0.0.0.0
      sel ecti on: 2
      ACMEPACKET(realm-config)#
      
```
4. Select the realm to which you want to apply this feature.
ACMEPACKET(realm-config)# **select**
i denti fier:
1: acmePacket <none> 192.168.20.0/24
2: realm1 <none> 0.0.0.0
sel ecti on: 2
ACMEPACKET(realm-config)#


```

      i denti fier:
      1: acmePacket <none>          192.168.20.0/24
      2: realm1      <none>          0.0.0.0
      sel ecti on: 2
      ACMEPACKET(realm-config)#
      
```
5. **trust-me**—Leave this parameter set to its default, **disabled**. This means that the realm is untrusted and with SIP PAI stripping enabled (which you will complete in a subsequent step of this procedure), the Net-Net SBC will strip all PAIs regardless of the privacy mode. The valid values are:
 - enabled | disabled
6. **pai-strip**—Enable PAI stripping. The default is **disabled**. Valid values are:
 - enabled | disabled

```
ACMEPACKET(real m-config)# pai -strip enabled
```

7. Save your work using the ACLI **save** or **done** command.

SIP Statuses to Q.850 Reasons

This section explains the Net-Net SBC's ability to map Q.850 cause values with SIP responses, a feature used in SIP calls and calls that require IWF.

RFC 3326 defines a header that might be included in any in-dialogue request. This reason header includes cause values that are defined as either a SIP response code or ITU-T Q.850 cause values. You can configure the 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.

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="compl eted el sewhere"
Reason: Q.850; cause=16; text="Termi nated"
```

This feature attends to the following possible SIP call scenarios:

- When the Net-Net SBC receives a SIP request or SIP response that contains the Reason header, the Net-Net SBC passes it without modification.
- When it generates a SIP response, the Net-Net SBC 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

Configuring reason cause mapping for SIP-SIP calls requires that you set up the ACLI local-response-map configuration with appropriate entries; these generate the SIP response and the Q.850 cause code value to be used for particular error scenarios. If you want to add a Reason header, then you need to enable that capability in the global SIP configuration.

To configure a local response map:

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

```
ACMEPACKET# config terminal
```

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

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

3. Type **local-response-map** and press <Enter>.

```
ACMEPACKET(session-router)# local-response-map
```

```
ACMEPACKET(local-response-map)#

```

4. Type entries and press <Enter>.

```
ACMEPACKET(local-response-map)# entries
```

```
ACMEPACKET(local-response-map-entry)#

```

From here, you can view the entire menu for the local response map entries configuration by typing a ?.

5. **local-error**—Set the local error that triggers the use of this local response map; there is no default for this parameter. Valid values are:
 - **invalid-message**—Response map for invalid messages
 - **cpu-overload**—Response map for CPU overload
 - **enum-void-route**—Response map for when an ENUM server returns a ENUM+VOID response, or the local route table has 0.0.0.0 as the next hop
 - **media-released**—Response map for media release conditions
 - **media-not-allocated**—Response map for when media is not allocated
6. **sip-status**—Set the SIP response code to use. There is no default and the valid range is:
 - Minimum—100
 - Maximum—699
7. **sip-reason**—Set the SIP reason string you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. **q850-cause**—Set the Q.850 cause. There is no default value.
9. **q850-reason**—Set the Q.850 reason string that you want to use for this mapping. There is no default value. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
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 SBC to add the Reason header:

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

```
ACMEPACKET# config terminal
```

2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
3. Type **sip-config** and press <Enter>.
ACMEPACKET(session-router)# **sip-config**
ACMEPACKET(sip-config) #
4. **add-reason-header**—Enable this parameter to add the Reason header. The default value is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration for changes to take effect.

Calls Requiring IWF

For interworking calls between SIP and H.323, you can configure:

- Mappings for SIP status codes to Q.850 values
- Mappings for particular Q.850 cause codes to SIP status codes

If it cannot find the appropriate mapping, then the 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 to 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
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

Q.850 Cause Value		SIP Status	Comments
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
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# **config terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
3. Type **sip-q850-map** and press <Enter>.
ACMEPACKET(session-router)# **sip-q850-map**
ACMEPACKET(sip-q850-map)#
4. Type **entries** and press <Enter>.
ACMEPACKET(sip-q850-map)# **entries**
ACMEPACKET(sip-q850-map-entry)#

From here, you can view the entire menu for the SIP status to Q.850 Reason with cause mapping entries configuration by typing a ?.
5. **sip-status**—Set the SIP response code that you want to map to a particular Q.850 cause code and reason. There is no default, and the valid range is:
 - Minimum—100
 - Maximum—699
6. **q850-cause**—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
7. **q850-reason**—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
8. Repeat this process to create the number of local response map entries that you need.
9. Save and activate your configuration for changes to take effect.

To configure a Q.850 cause to a SIP status with reason mapping:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config 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 is:
 - Minimum—100
 - Maximum—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.

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**—Enable this parameter to add the Reason header. The default is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration for changes to take effect.

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-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 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 trgp
- 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.

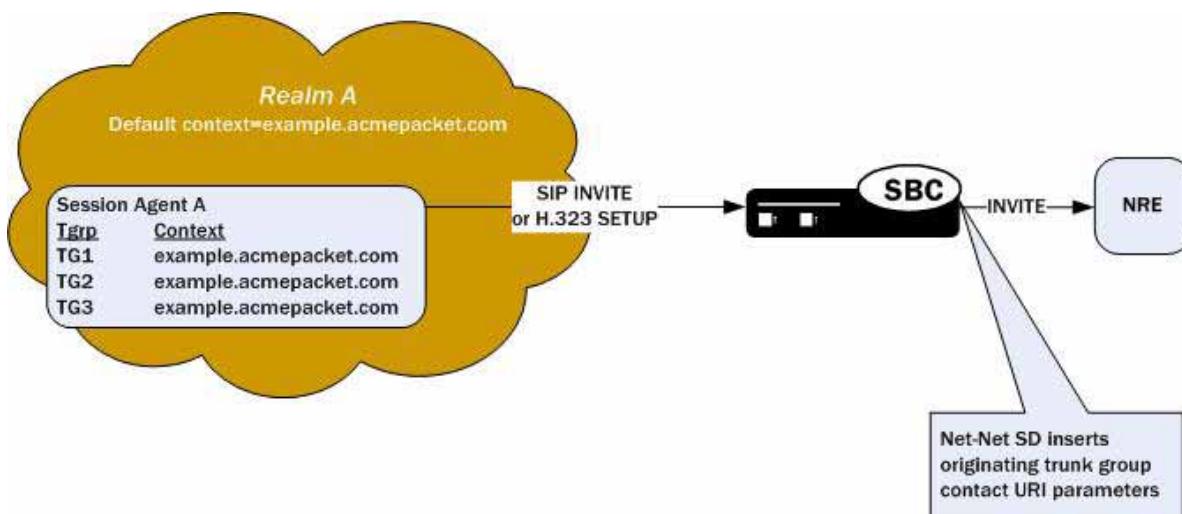
In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net SBC allows you to define URI parameters for use with originating and terminating trunk group URIs.

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 (trgp) 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: trgp (identifier of the specific trunk group) and trunk-context (defines the network domain of the trunk group). These appear in the following formats:
 - trgp="trunk group name"
 - trunk-context="network domain"

The URI BNF for would appear as it does in the example directly below, where the trgp is tg55 and the trunk-context is trunk-context=tel co. exempl e. com:

tel : +15555551212; trgp=tg55; trunk-context=tel co. exempl e. com

2. The second format is customized specifically for access URIs and contains two provisioned parameters: trgp (or tgname) and context (or provstring). This appears as trgp.context, where these definitions apply:
 - trgp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
 - context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

custsi te2NY-00020. type2. voi p. carrier. net

where the trgp is custsi te2NY-00020, and the context is type2. voi p. carrier. net.

The BNF for an access URI conforms to the following:

```

SIP-URI = "sip: " [userinfo] hostport uri-parameters [headers]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param

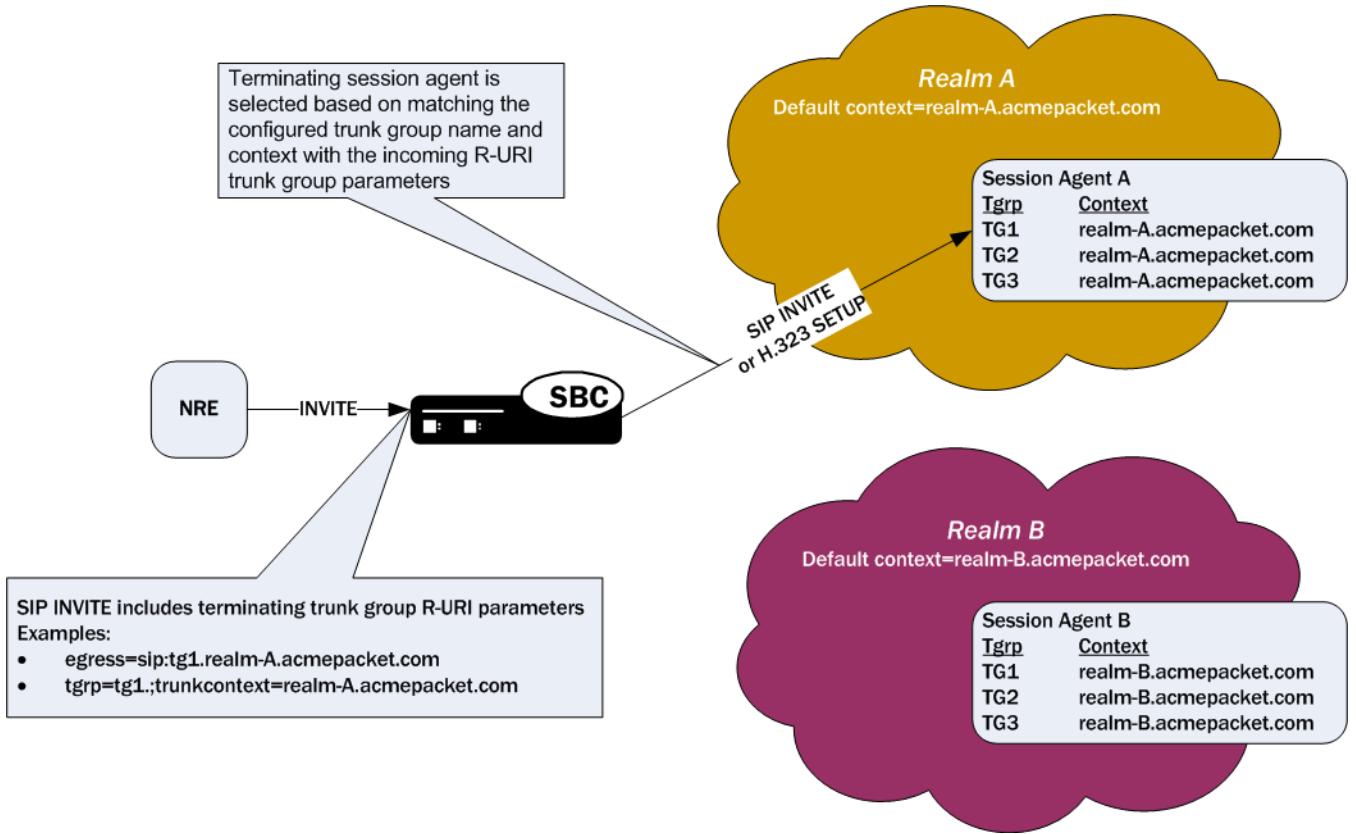
other-param = accessid / pname [ '=' pvalue ]
accessid = "access=" accessURI

accessURI = scheme tgname [". " provstring]
scheme = "sip: " / token
tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
provstring = *(domain ". ") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *( alphanum / "-" ) alphanum
domain = alphanum/ alphanum *( alphanum / "-" ) alphanum

```

Terminating Trunk Group URI Parameters and Formats

Terminating trunk group URI parameters appear in the R-URI, and they can be included in by a network routing element to instruct the 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 i sp. exempl e. net@egwy. i sp. exempl e. net:

```
I INVITE si p: +15555551212; tgrp=TG2-1; trunk-
context=i sp. exempl e. net@egwy. i sp. exempl e. net SIP/2.0
```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: trgp (or tgname) and context (or tgdomain). This appears as trgp.context (or tgname.tgdomain), where definitions apply:
 - tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character

- context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```

SIP-URI = "sip: " [userinfo] hostport url-parameters [headers]
url-parameters = *( ";" url-parameter )
url-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 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 “trgp” 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-SIP calls, you set the **preserve-user-info option** in the SIP interface configuration.
- For SIP-H.323 calls requiring IWF, you set the **preserve-user-info-sa option** in the session agent configuration.

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-trgp-mode**—Set the mode that you want to use for routing for terminating trunk group URIs. The default is **none**. Your choices are:
 - **none**—Disables routing based on trunk groups
 - **iptel**—Uses trunk group URI routing based on the IPTEL formats
 - **egress-uri**—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **session-agent** and press <Enter>.
ACMEPACKET(session-router)# **session-agent**
ACMEPACKET(session-agent)#
 4. **out-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the 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: `trgp: context` or `trgp. 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.

Your ACLI entries for this list must take one of these formats: `trgp: context` or `trgp. 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(real m-config)#
4. **trunk-context**—Enter the trunk group context to use for this realm. There is no default.

Using this Feature with a SIP Interface

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the **preserve-user-info** option for that SIP interface.

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **session-group** and press <Enter>.
ACMEPACKET(session-router)# **sip-interface**
ACMEPACKET(sip-interface)#
 4. **options**—Add support for trunk group URIs with SIP interface that uses registration caching.

If you are adding this option to a new SIP interface, you can just type **options**, a <Space>, and **preserve-user-info**.

If are adding this to an existing SIP interface, you must type a plus (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +preserve-user-info**.

Example 1: Adding Originating Trunk Group Parameters in IPTEL Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTEL format.

```

sip-manipulation
  name          add_iptel
  header-rule
    name        contact
    action      manipulate
    match-value
    msg-type   any
  
```

```

element-rule
  name          tgrp
  type          uri-user-param
  action        add
  match-val-type
  match-value
  new-value      any
                           $TRUNK_GROUP

element-rule
  name          trunk-context
  type          uri-user-param
  action        add
  match-val-type
  match-value
  new-value      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          add_att
  header-rule
    name          contact
    action        manipulate
    match-value
    msg-type     any
  element-rule
    name          egressURI
    type          uri-param
    action        add
    match-val-type
    match-value
    new-value      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          strip_iptel
  header-rule
    name          request-uri
    action        manipulate
    match-value
    msg-type     any
  element-rule
    name          tgrp
    type          uri-user-param
    action        delete-element
    match-val-type
    match-value
    new-value      any
  element-rule
    name          trunk-context
    type          uri-user-param
    action        delete-element

```

match-val -type	any
match-val ue	
new-val ue	

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

```

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.

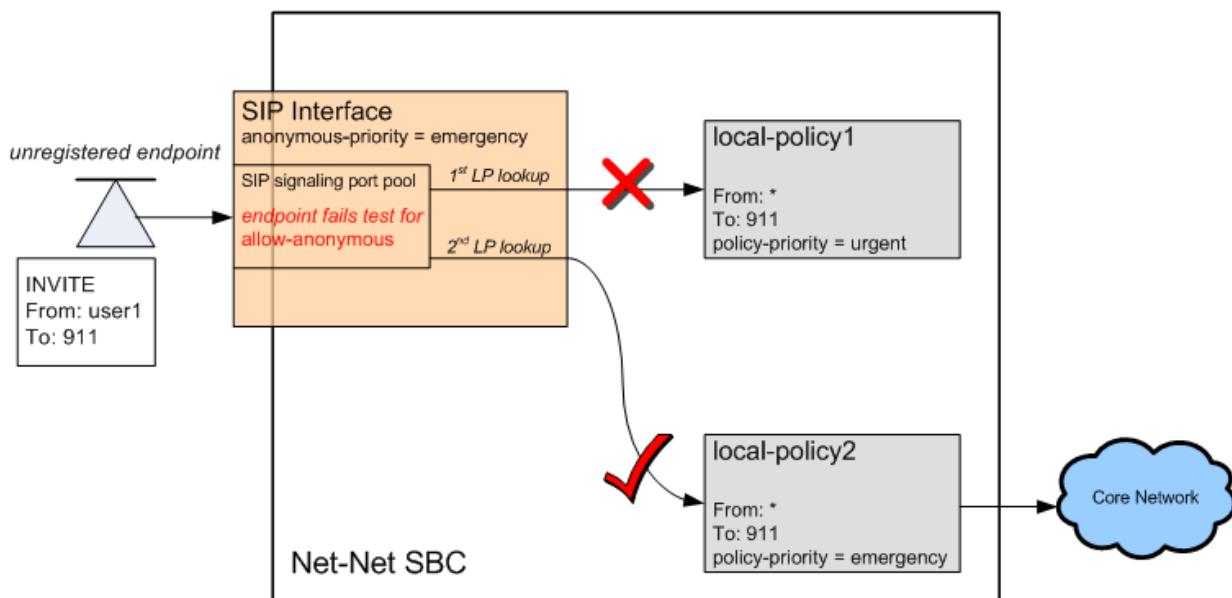
When the Net-Net SBC receives a non-allowed emergency request, it performs a local policy lookup for a matching local policy. An emergency local policy could be configured to match if the To: header in a SIP message was addressed to 911.

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

- ACMEPACKET(sip-interface) #
4. Type **select** and the number of the SIP interface you want to configure.
- ```
ACMEPACKET(sip-interface) # select 1
```
5. **anonymous-priority**—Set the policy priority for this SIP interface. It is used to facilitate emergency sessions from unregistered endpoints. This value is compared against the **policy-priority** parameter in the **local-policy** configuration element. The default is **none**. The valid values are:
    - none | normal | non-urgent | urgent | emergency
- This completes the configuration.
- ```
ACMEPACKET(sip-interface) # anonymous-priority emergency
```
6. Save your work using the ACLI **done** command.

Setting Policy Priority

To set the policy priority for a local policy:

1. In Superuser mode, type **configure terminal** and press <Enter>.
- ```
ACMEPACKET# configure terminal
```
2. Type **session-router** and press <Enter> to access the session-level configuration elements.
- ```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router) #
```
3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
- ```
ACMEPACKET(session-router) # local-policy
ACMEPACKET(local-policy) #
```
4. Type **select** and the number of the local policy you want to configure.
- ```
ACMEPACKET(local-policy) # select 1
```
5. **policy-priority**—Enter the policy priority for this local policy. It is used to facilitate emergency sessions from unregistered endpoints. This value is compared against the **anonymous-priority** parameter in the **sip-interface** configuration element. The default is **none**. The valid values are:
 - none | normal | non-urgent | urgent | emergency
- This completes the configuration.
- ```
ACMEPACKET(local-policy) # anonymous-priority emergency
```
6. Save your work using the ACLI **done** command.

## Fraud Prevention

---

The Net-Net SBC 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:

1. In Superuser mode, type **configure terminal** and press <Enter>.
- ```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

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

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

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

```
4. Type **select** to configure the existing sip config.

```
ACMEPACKET(sip-config)# select

```
5. **sip-message-len**—Set the size constraint in bytes of a SIP message. The default is **4096**. The valid range is:
 - Minimum—0
 - Maximum—65535
This completes the configuration.

```
ACMEPACKET(sip-config)# sip-message-len 5000

```
6. Save your work using the ACLI **done** command.

SIP Early Media Suppression

This section explains how to configure SIP early media suppression, which lets you determine who can send early media and in what direction. Early media are the RTP/RTCP packets sent from the called party to the caller, or vice versa, before a session is fully established (before a 200 OK is received). When the 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

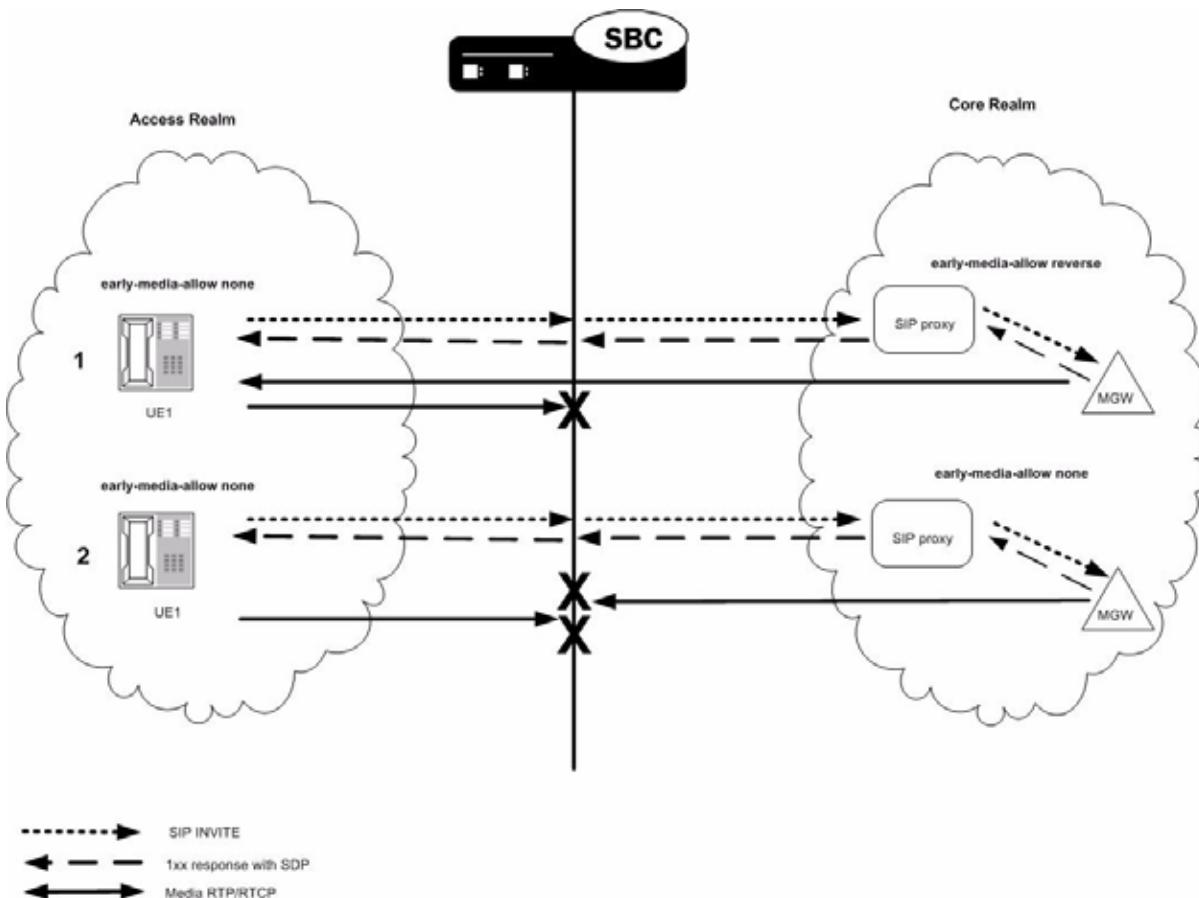
parameter value then applies to either allow all, block all, or block one-way early media until a 200 ok is received. At that point bidirectional media is allowed. The decision is based on SIP-layer addressing of next-hops.

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 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# config terminal
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# media-manager
3. Type **realm** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# realm
ACMEPACKET(realm)#
4. If configuring an existing realm, enter the **select** command to select the realm.
5. **early-media-allow**—Enter the early media suppression rule for the realm. The valid values are:

- **none**—No early media is allowed in either direction
- **both**—Early media is allowed in both directions
- **reverse**—Early media received by 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	
identity	access1
addr-prefix	192.168.1.0/24
network-interfaces	
media:0	
mm-in-real-m	enabled
mm-in-network	enabled
msm-relax	disabled
qos-enabled	disabled
max-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observe-window-size	0
parentrealm	
dnsrealm	
mediapolicy	
in-translational	
out-translational	
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)#
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-ip	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	
redirection-action	
loose-routing	enabled
send-media-session	enabled
response-map	
ping-method	
ping-interval	0
media-profiles	
in-translations	
out-translations	
trust-me	disabled
early-media-allow	reverse

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(real-m-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(real-m-group)# source-realm "Private, Public"
```

To add a realm to the list, use the plus sign (+) in front of each new entry.

```
ACMEPACKET(real-m-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(real-m-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:

```
ACMEPACKET(real-m-group)# source-realm "Private, Public"
```

To add a realm to the list, use the plus sign (+) in front of each new entry.

```
ACMEPACKET(real-m-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(real-m-group)# destination-realm -Private
```
8. **early-media-allow-direction**—Set the direction for which early media is allowed for this realm group. Valid values are:
 - **none**—Turns off the feature for this realm group by blocking early media
 - **reverse**—Allows early media to flow from called to caller
 - **both** (default)—Allows early media to flow to/from called and caller
9. Save and activate your configuration.

SDP-Response Early Media Suppression

This section explains how to configure SDP-response early media suppression, which can be used when the Net-Net SBC is deployed after a softswitch or proxy in the signaling path. In this deployment, user endpoints and gateways communicate directly with the softswitch or proxy, which in turn sends call signaling to the Net-Net SBC. The call signaling gets sent back to the same or different softswitch or proxy. Because the Net-Net SBC does not communicate with the endpoints or gateways that are the media terminators, early media suppression for this deployment must use SDP-based addressing rather than the SIP-based addressing (described in the *SIP Early Media Suppression* section in this technical notice).

Using this feature lets you configure specific IP addresses for which early media should not be suppressed, based on SDP addressing. The Net-Net SBC checks the SDP addresses in SIP responses against these IP address or address ranges to determine on which media gateway a call terminates.

How it Works for SIP-Based Addressing

With the SIP-based addressing described in the *SIP Early Media Suppression* section, early media suppression is based on the outbound SIP interface realms and the value of their early-media-allow parameter. When the 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.

How it Works with SDP-Based Addressing

SDP-response early media suppression follows the same sequence described for SIP-based addressing with one exception. A provisional response with SDP media can make the Net-Net SBC select a new early-media-allow rule from another realm, based on the addressing inside the responding SDP.

When the SDP-response early media suppression feature is enabled, the Net-Net SBC searches the outbound SIP interface's realms for a matching address prefix with the connection address in the responding SDP. If it finds a match, it uses the early-media-allow parameter value of that realm until the 200 OK message is received, then bidirectional media is allowed regardless. If the Net-Net SBC does not find a match, it searches all of the global realms for one. If it finds a match, the Net-Net SBC uses that realm's early-media-allow parameter value. If it does not find a match in the global realm(s), the Net-Net SBC continues to use the previous early-media-allow parameter value.

Global Realms

Global realms are realms that are not parents or children of any other realms, do not have defined SIP interfaces and ports (or any signaling interface or stack), and are configured to use the network interface lo0:0. They are special realms, applicable system-wide, and are currently only used for this feature. The only global realm configuration parameters applicable to early media suppression are:

- addr-prefix
- additional-prefixes
- early-media-allow

- network-interface (which must be set to lo0:0)

Additional Prefixes

You can specific additional prefixes in addition to that of the addr-prefix parameter you configure for a realm. For example, you can configure a global realm with additional address prefixes to specify the IP addresses (or ranges of addresses) of the media gateways that are allowed to send and receive early media. This overrides the SIP interface realm's early media blocking settings.

You can also enter additional prefixes in non-global realms. These additional prefixes function the same as would multiple values in the addr-prefix parameter (which only takes one value), except addresses in additional-prefixes are not used for SIP NATs.

Using the SDP-Response Early Media Suppression Rule

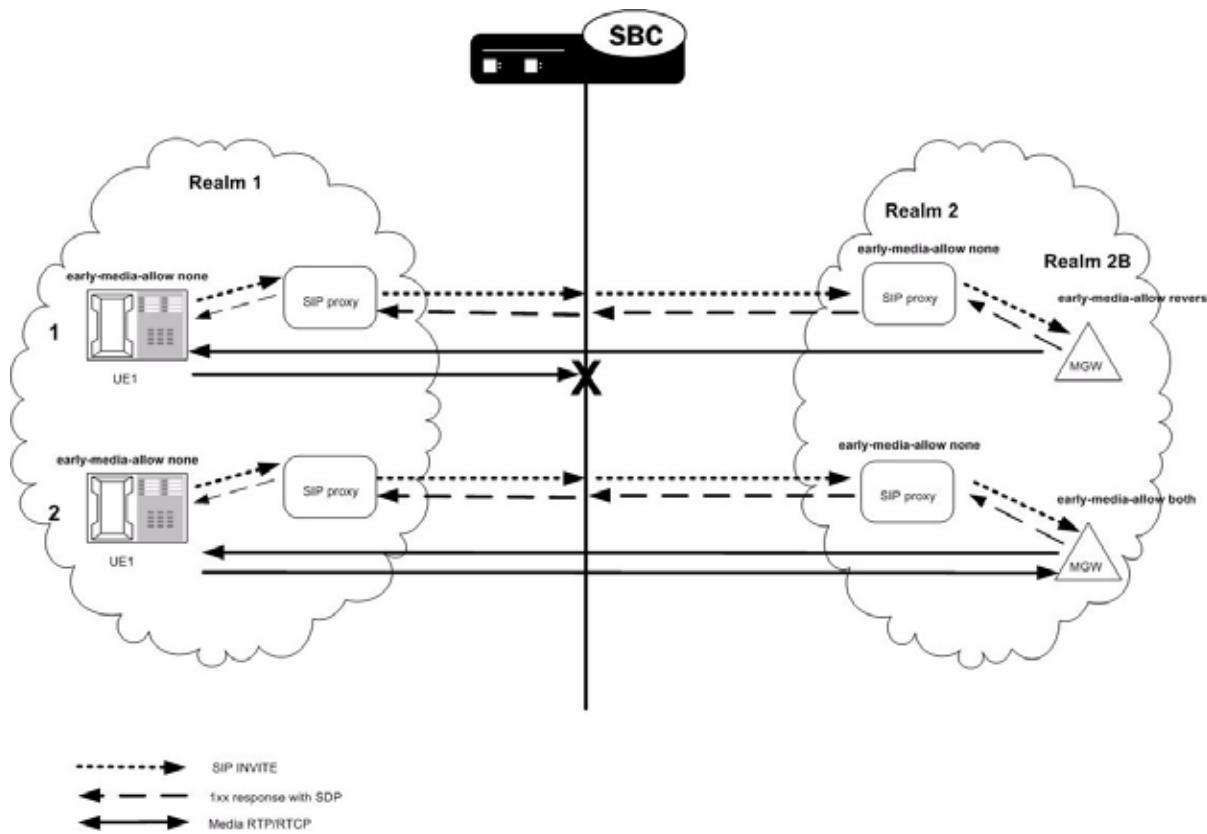
To use SDP-response early media suppression, you must add the `early-media-sdp-realms` option to the SIP interface configuration that interfaces with the next-hop device, such as the supported softswitch.

When the Net-Net SBC receives a provisional response that includes SDP from the called endpoint, and the `early-media-sdp-realms` option is active in the outgoing SIP interface of the call, it first searches the realms that apply to the outgoing SIP interface. If it does not find a realm, the Net-Net SBC searches the global realms. If the search yields a new realm that is not the SIP interface realm, its early media suppression rule (if any) replaces the existing one. Only the early media suppression rule of the new realm is applied to the call. Other realm properties from the outbound realm remain applicable to the call. If no new realm is found, the early media policy of the outgoing SIP interface realm is applied.

The Net-Net SBC allows media when the SDP media connect address in a response matches one of a configured list of IP address ranges defined in a realm and the realm has early media allowed. You need to configure specific a IP address or address range to specify which media gateways should not be suppressed based on SDP media addresses. The IP addresses are checked against the SDP being received. The decision for suppression is based on whether the matching realm allows early media. The early media will be suppressed if the matching realm does not allow early media or if there is no match and the outbound SIP interface realm does not allow early media.

Example

The following illustration shows two examples of SDP-response early media suppression.



Configuring SDP-Response Early Media Suppression

To configure SDP-response early media suppression:

1. Add the `early-media-sdp-realms` option to the SIP interface that interfaces with the softswitch.
2. Configure the SIP interface realm with an early media suppression rule that blocks all early media.
3. Configure either or both of the following:
 - One or more of the SIP realm's child realms, each with an early media suppression rule that allows all or reverse direction early media and a list of additional prefixes that specifies the IP addresses of the media gateways, or a range of IP addresses that includes the media gateways. Early media is allowed from these gateways only for calls that signal through this SIP interface.
 - One or more realms that has the network interface equal to `loop:0`, an early media suppression rule that allows all or reverse direction early media and a list of additional prefixes that specifies the IP addresses of the media gateways, or a range of IP addresses that includes the media gateways. Early media is allowed from these gateways regardless of interface.

Configuring the SIP Interface

To configure a SIP interface:

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

From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.
4. If configuring an existing interface, enter the **select** command to select the interface.
5. **options**—Enter early-media-sdp-realms as the option. If adding to an existing list of options, use a preceding plus (+) sign.
options +early-media-sdp-realms
6. Continue to the next section to configure the outbound realm.

For example:

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)# options +early-media-sdp-realms
ACMEPACKET(sip-interface)# done
sip-interface
  state          enabled
  realm-id      access1
  sip-port
    address      192.168.1.30
    port         5060
    transport-protocol  UDP
    allow-anonymous  all
  carriers
    proxy-mode   Proxy
    redirect-action
    contact-mode
    nat-traversal
    nat-interval
    registration-caching
    min-reg-expire
    registration-interval
    route-to-Registrar
    tel-uri-scheme
    uri-fqdn-domain
    options
    trust-mode
    last-modified-date 2006-05-10 18:27:31
```

Configuring a Realm**To configure a realm:**

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.
ACMEPACKET# **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(real m-config)#
 4. If configuring an existing realm, enter the **select** command to select the realm.
 5. **early-media-allow**—Enter the early media suppression rule for the realm. The valid values are:
 - **both**—Early media is allowed in both directions
 - **reverse**—Early media received by Net-Net SBC in the reverse direction is allowed
 - **none**—Early media is blocked
 6. **additional-prefixes**—Enter a single or a comma-delimited list of IP address prefixes to use in addition to the value of the **addr-prefix** parameter.

<IPv4> [<number of bits>]

<IPv4> is a valid IPv4 address and <number of bits> is the number of bits to use to match an IP address with the address prefix. Not specifying <number of bits> implies that all 32 bits are used for matching.

Enclose the list between quotes if there is any space between a comma and the next address prefix.

You can add and remove address prefixes to and from the list:

- **add-additional-prefixes** adds one or more additional prefixes
add-additional-prefixes 192.168.201.69
- **remove-additional-prefixes** removes one or more additional prefixes
remove-additional-prefixes 192.168.201.69

If using multiple address prefixes, enter a comma-delimited list.

7. Save and activate your configuration.

For example:

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(real m-config)# additional-prefixes
192.168.200.0/24, 192.168.201.68
ACMEPACKET(real m-config)# done
real m-config
  identifier          early-media
  addr-prefix        0.0.0.0
  network-interfaces

  mm-in-realm        media2:0
  mm-in-network      disabled
  enabled
```

msm-rel ease	di sabl ed
qos-enabl e	di sabl ed
max-bandwi dth	0
max-l atency	0
max-j i tter	0
max-packet-l oss	0
observ-wi ndow-si ze	0
parent-real m	
dns-real m	
medi a-pol i cy	
i n-transl ati oni d	
out-transl ati oni d	
i n-mani pul ati oni d	
out-mani pul ati oni d	
cl ass-profi le	
average-rate-l imi t	0
access-control -trust-l evel	none
i nval i d-si gnal -threshol d	0
maxi mum-si gnal -threshol d	0
untrusted-si gnal -threshol d	0
deny-peri od	30
symmetri c-l atchini g	di sabl ed
pai -strip	di sabl ed
trunk-context	
earl y-medi a-al low	reverse
addi ti onal -prefi xes	192. 168. 200. 0/24
	192. 168. 201. 69
Last-modifi ed-date	2006-05-11 06: 47: 31

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 local response code map entries.

To apply a new status code to a SDP address correlation mismatch:

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

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

```

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

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

```

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

```
ACMEPACKET(session-router)# local-response-map
ACMEPACKET(local-response-map)#

```

4. Type **entries** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(local-response-map)# entries
ACMEPACKET(local-response-map-entry)#

```

5. **local-error**—Enter **sdp-address-mismatch** for which to apply the new status code.

6. **sip-status**—Enter the new status code you want to use.

7. **sip-reason**—Enter the reason to correlate with the new status code.

```
ACMEPACKET(local-response-map-entry)# local-error sdp-address-
mismatch
ACMEPACKET(local-response-map-entry)# sip-status 403
ACMEPACKET(local-response-map-entry)# sip-reason sdp address mismatch
```

8. Save and activate your configuration.

In addition, note that you apply this feature per-realm by setting the enforcement profile 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(medi a-manager) #
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
- ```
ACMEPACKET(medi a-manager) # realm-config
ACMEPACKET(real m-config) #
```
4. **enforcement-profile**—Enter the name of the enforcement profile you want to apply to this realm.
- ```
ACMEPACKET(real m-config) # enforcement-profile profile1
```
5. Save and activate your configuration.

SDP Insertion for (Re)INVITEs

If your network contains some SIP endpoints that do not send SDP in ReINVITEs but also contains others that refuse INVITEs without SDP, this feature can facilitate communication between the two types. The 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.

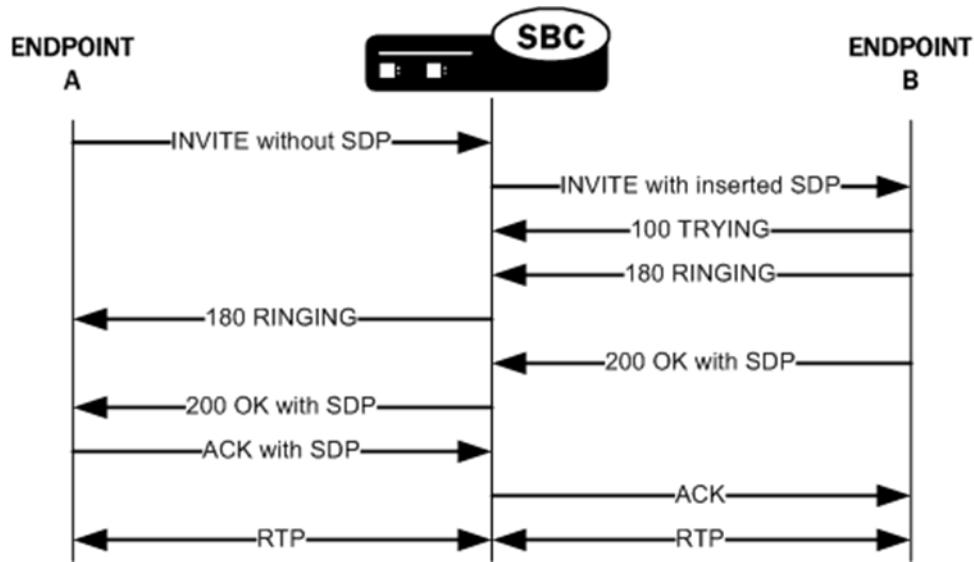
This feature works for both INVITEs and ReINVITEs.

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.

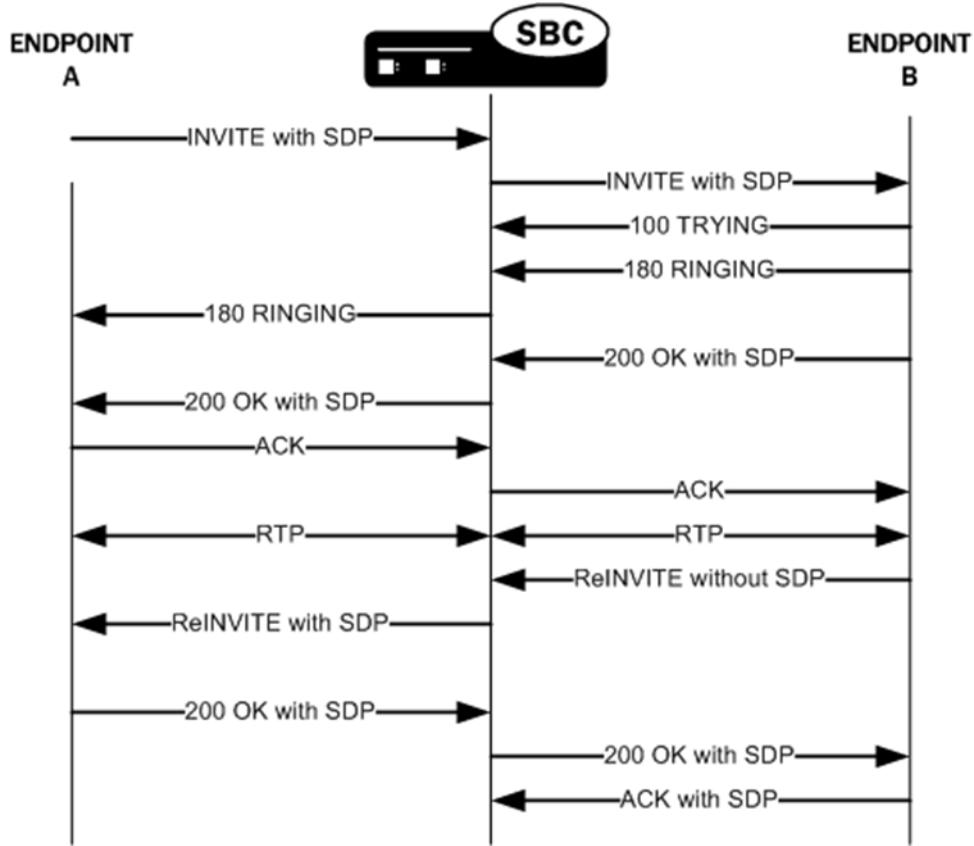
SDP Insertion for SIP INVITES

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.



ACLI Instructions and Examples

Configuring SDP Insertion for SIP INVITES

This section shows you how to configure SDP insertion for the calls cases described above.

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

```
ACMEPACKET(config)# 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. It sets the source address and address prefix bits in the flow. It also parses and loads the source flow address into the MIBOCO messages. After receiving the calling SDP, the 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. In addition, sipd sends the information for both the incoming and outgoing flows. After receiving SDP from the called side sipd, the Net-Net SBC sends information for both flows to the MBCD so that the correct NAT entries are installed.

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).

Configuring Restricted Latching

ACLI Instructions and Examples

You can configure restricted latching using the ACLI.

To configure restricted latching:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
 2. Type **media-manager** and press <Enter> to access the media-level configuration elements.
ACMEPACKET(config)# 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(real m-config)#
 4. Select the realm where you want to apply this feature.

```
ACMEPACKET(real m-config)#
  identifier:
    1: Acme_Real m <none>          0. 0. 0. 0
    2: MGCP_Real m <none>          0. 0. 0. 0
    3: H323REALM <none>           0. 0. 0. 0
```

```
selection:
ACMEPACKET(real m-config)#
```

5. **restricted-latching**—Enter the restricted latching mode. The default is **none**. The valid values are:
 - **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. This field will be used only if the restricted-latching is used. The default is **32**; if this parameter uses this value, the complete IP address is matched. The valid range is:
 - Minimum—1
 - Maximum—32
7. Save your work using the ACLI **done** and **save** commands.

The following example shows the realm configuration.

```
real m-config
  identifier          Acme_Real m
  addr-prefix        0. 0. 0. 0
  network-interfaces
    public: 0
    mm-in-real m   enabled
    mm-in-network  enabled
    msrn-release   disabled
    qos-enabled    disabled
    max-bandwidth  0
    max-latency    0
    max-jitter     0
    max-packet-loss 0
    observ-windown-size 0
    parent-realm
    dns-realm
    media-policy
    in-transl-ationd
    out-transl-ationd
    class-profile
    average-rate-limit 0
    access-control-trust-level
    invalid-signal-threshold 0
    maximum-signal-threshold 0
    deny-period    30
    symmetric-latching  disabled
    pair-strip     enabled
    mm-in-system   enabled
    restricted-latching sdp
```

restriction-mask	30
Last-modified-date	2006-05-20 12:49:43

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(config)# **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.

Dynamic Transport Protocol Change

The Net-Net SBC also uses the IP address and port in the Contact and Via headers. This is useful for cases when endpoints dynamically change transport protocols (TCP/UDP), and the port number used for sending an INVITE might not be the same one used to send a Register message.

If you do not enable this feature, when an endpoint registered with the Net-Net SBC used UDP for its transport protocol, a call fails if that endpoint subsequently initiates the call using TCP. The Net-Net SBC checks for the Layer 3 IP address and port, and it rejects the call if the port is changed.

With the new option `reg-no-port-match` added to the SIP interface configuration, the Net-Net SBC will not check the Layer 3 port in the INVITE and REGISTER messages.

ACLI Instructions and Examples

To enable dynamic transport protocol change:

1. In Superuser mode, type `configure terminal` and press <Enter>.
ACMEPACKET# **config 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)#[/b]
4. Type `options +reg-no-port-match` and press <Enter>.
ACMEPACKET(sip-interface)# **options +reg-no-port-match**
If you type `options reg-no-port-match` without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.
5. Save and activate your configuration.

SIP Privacy Extensions

This section explains how you can configure privacy services to be applied only when the source is trusted and the destination is considered untrusted. (Prior to this release, the Net-Net SBC always applied the privacy services, unless the source and the destination were both trusted.)

The Net-Net SBC considers all user endpoints and nodes outside the core as untrusted.

How it Works

The Net-Net SBC acts as the boundary device between the trusted platform and the untrusted Internet, to implement privacy requirements. When it receives a message, the Net-Net SBC checks whether the source is trusted. It evaluates the level of privacy requested in a Privacy header, if present.

Depending on whether the source is trusted or untrusted, the Net-Net SBC can do different things when passing the message to the outgoing side. It also checks whether the destination is trusted.

Privacy Types Supported

The Net-Net SBC supports the following Privacy types:

- user: user-level privacy function provided. Any non-essential informational headers are removed, including the Subject, Call-Info, Organization, User-Agent, Reply-To, and In-Reply-To. Possibly the original value of the From header is changed to anonymous.
- header: headers that cannot be set arbitrarily by the user (Contact/Via) are modified. No unnecessary headers that might reveal personal information about the originator of the request are added. (The values modified must be recoverable when further messages in the dialog need to be routed to the originator.)
- id: third-party asserted identity kept private with respect to SIP entities outside the trust domain with which the user authenticated.

The following SIP headers can directly or indirectly reveal identity information about the originator of a message: From, Contact, Reply-To, Via, Call-Info, User-Agent, Organization, Server, Subject, Call-ID, In-Reply-To and Warning.

user

The Net-Net SBC supports the Privacy type user. It can remove non-essential information headers that reveal user information by:

- Setting the SIP From header and display information to anonymous
- Removing the Privacy header
- Removing Proxy-Require option tag = privacy (if present)
- Removing the following headers:
 - Subject
 - Call-Info
 - Organization
 - User-Agent
 - Reply-To
 - In-Reply-To

header

The Net-Net SBC also supports the Privacy type header. It modifies SIP headers that might reveal the user identity by:

- Stripping the Via header
- Replacing the Contact header
- Stripping Record-Route
- Removing the Privacy header
- Removing Proxy-Require option tag = privacy (if present)

In general, the B2BUA behavior of the Net-Net SBC by default provides header privacy for all sessions.

id

The Net-Net SBC also supports the Privacy type id. It keeps the Network Asserted Identity private from SIP entities outside the trusted domain by:

- Stripping only P-Asserted-Identity
- Removing the Privacy header and Proxy-Require option-tag = privacy
- Setting the From header to anonymous (for the backward compatibility)

Examples

The following examples show the actions the Net-Net SBC performs depending on the source and target of the calls.

Calls from Untrusted Source to Trusted Target

When calls are from an untrusted source to a trusted target and PPI is included in the INVITE sent to IP border elements, the Net-Net SBC maps the PPI information to PAI in the outgoing INVITE to the trusted side (even if the Privacy header is set to id or to none). The Privacy and From headers get passed on unchanged.

IP border elements must pass PAI (if received in the ingress INVITE) and the From and Privacy headers to the egress side just as they were received on the ingress side.

The Net-Net SBC maps the PPI to PAI by default, if the outgoing side is trusted. To change this behavior, you need to configure the disable-ppi-to-pai option.

Calls from Trusted to Untrusted

When calls are from a trusted source to an untrusted target, and the Privacy header is set to id, the Net-Net strips PAI, makes the From header anonymous, and strips the Privacy header.

If the Privacy header is set to none, the Net-Net SBC does not change the From header and passes on the Privacy header, if there is one.

Calls from Trusted to Trusted

When calls are going from trusted source to trusted target acting as a peer network border element and PPI is included, the Net-Net SBC maps PPI to PAI. The Privacy header remains the same as signaled and the Net-Net SBC passes the From header and the PAI without changes.

Configuring SIP Privacy Extensions

Prior to this release the session agent's trust mode provided this functionality. Now you configure SIP interface's trust-mode as none, which means nothing is trusted for this SIP interface.

You also configure the disable-ppi-to-pai parameter disable the changing of the P-Preferred header to the P-Asserted-Identity header, if the outgoing side is trusted.

Trust Mode**To configure the trust mode:**

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

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

```

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

4. If configuring an existing interface, enter the select command to select the interface.
5. **trust-mode**—Select the trust mode for this SIP interface. The default value is **all**. The valid values are:
 - **all**—Trust all previous and next hops except untrusted session agents
 - **agents-only**—Trust only trusted session agents
 - **realm-prefix**—Trusted only trusted session agents or address matching realm prefix
 - **registered**—Trust only trusted session agents or registered endpoints
 - **none**—Trust nothing
6. Save and activate your configuration.

The following example shows the trust-mode set to none. The remaining SIP interface options are omitted for brevity.

```
sip-interface
  state          enabled
  real-m-id      access1
  sip-port
    address      192.168.1.30
    port         5060
    transport-protocol UDP
    allow-anonymous all

  carriers
    proxy-mode   Proxy
    redirect-action
    contact-mode maddr
    nat-traversal none
    nat-interval 30
    registration-caching disabled
    min-reg-expire 300
    registration-interval 3600
    route-to-registry disabled
    tel-uri-scheme disabled
    uri-fqdn-domain
    options
    trust-mode    none
```

Disabling the PPI to PAI Change

To disable the changing of PPI to PAI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
`ACMEPACKET# configure terminal`
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
`ACMEPACKET(configure)# session-router`
3. Type **sip-config** and press <Enter>. The system prompt changes.

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

```

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

4. If configuring an existing SIP configuration, enter the select command to select it.
5. **options**—Enter **disable-ppi-to-pai**. If adding to an existing list of options, use a preceding plus (+) sign.
options +disablene-ppi-to-pai
6. Save and activate your configuration.

SIP Registration Cache Limiting

Using SIP registration cache limiting for SIP endpoint access deployments, you can restrict the size of the SIP registration cache for the global SIP configuration.

You can implement this feature if you have been seeing issues where, either due to network failure scenarios or incorrect sizing of system capabilities, the 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 modification), 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.

Monitoring Information

You can monitor how many entries are in the SIP registration cache using the ACLI **show registration** command and referring to the Local Contacts statistics.

ACLI Instructions and Examples

This section shows you how to configure the registration cache limit, and how to set the options controlling retry times and thresholds for alarm purposes.

To configure SIP registration cache limiting:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

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

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

```

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

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

```

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

4. **registration-cache-limit**—Set the registration cache limit, or the maximum number of SIP registrations that you want to keep in the registration cache. The minimum and default value for this parameter is 0, and you can set it to a maximum value of 999999999. Leaving this parameter set to 0 means there is no limit on the registration cache (and therefore leaves this feature disabled).
5. **options**—Set the options parameter by typing options, a <Space>, the option name **reg-cache-lim-retry-after=X** (where X is the value added to the Retry-After header) with a “plus” sign in front of it. This option defaults to 1800, and you can enter values from 0 to 999999999.

You can configure the alarm threshold option the same way, substituting the option name **reg-cache-alarm-thresh=X** (where X is the percentage of registration cache limit that triggers an alarm). This option defaults to 95, and you can enter value from 0 to 100.

```
ACMEPACKET(sip-config)# options +reg-cache-lim-retry-after=2500
ACMEPACKET(sip-config)# options +reg-cache-alarm-thresh=90
```

If you type options and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

SIP Registration Overload Protection

You can configure your Net-Net SBC for SIP Registration overload protection, which augments the Net-Net SBC’s 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 forwards the REGISTER request to the registrar. To challenge the endpoint’s registration, the registrar sends back either a 401 Unauthorized or 407 Proxy Authorization Required response. When it receives the 401 or 407, the 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 retransmissions 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 able to find the temporary registration cache and the saved challenge context when the second REGISTER arrives, it forwards the REGISTER without checking the maximum registration sustain rate constraint for ingress and egress session agents—thereby ensuring that the REGISTER with authentication credentials is sent to the registrar. So when you use this feature, you should set the maximum registration sustain rate constraint of the session agent (representing the registrar) at half the registrar's maximum registration sustain rate. Additional REGISTER requests with the same challenge context are subject to the maximum registration sustain rate constraint.

ACLI Instructions and Examples

When you configure this feature, be sure to set the **reg-overload-protect** option in your global SIP configuration:

To enable SIP Registration overload protection on your Net-Net SBC:

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

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# config terminal
ACMEPACKET(config)#
```

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

```
ACMEPACKET(config)# 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# config terminal
ACMEPACKET(config)#

```

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

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

```

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

```
ACMEPACKET(session-router)# sessi on-constrai nts
ACMEPACKET(session-constrai nts)#

```

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-constrai nts)# rate-constra lnts
ACMEPACKET(rate-constra lnts)#

```

5. **method**—Enter the SIP method name for the method you want to throttle. Although the parameter accepts other values, your entries should come only from the from the following list for the feature to function properly:

- NOTIFY
- OPTIONS
- MESSAGE
- PUBLISH
- REGISTER

6. **max-inbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
7. **max-outbound-burst-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound burst rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
8. **max-inbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the inbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
9. **max-outbound-sustain-rate**—For the SIP method you set in the methods parameter, enter the number to restrict the outbound sustain rate on the SIP interface where you apply these constraints. The default and minimum value is 0, and the maximum is 999999999.
10. Save your changes and apply this session constraint and its rate constraint(s) to SIP interfaces.

Applying Session and Rate Constraints to a SIP Interface

You need the name of the session constraints configuration to apply the restrictions you set up to a SIP interface.

To apply session and rate constraints to a SIP interface:

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

```
ACMEPACKET# config ure termi nal
ACMEPACKET(configure)#

```

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

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

```

3. Type **sip-interface** and press <Enter>.

```
ACMEPACKET(session-router)# si p-i nterface
ACMEPACKET(sip-interface)#

```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **constraint-name**—Enter the name of the session constraint configuration where you have set up rate constraints to apply them to this SIP interface. This parameter has no default, and must be the valid name of a session constraint configuration.
5. Save and activate your configuration.

Configuring Rate Constraints for Session Agents

You can also use this feature for individual SIP session agents.

To configure rate constraints for a SIP session agent:

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

```
ACMEPACKET# config terminal  
ACMEPACKET(config)#
```

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

```
ACMEPACKET(config)# 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 99999999.

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 (which is 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, 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 Instruction and Examples

You enable this feature on a per-realm basis by setting one parameter.

To enable SIP delayed media update:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter> to access the signaling-related configurations.
ACMEPACKET(configure)# **media-manager**
3. Type **realm-config** and press <Enter>.
ACMEPACKET(media-manager)# **realm-config**
If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **delay-media-update**—Enable keeping the Net-Net SBC from updating its media flow information for flows established after an offer/answer exchange. The default is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

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 used, 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 \(455\)](#) section of this document for configuration prerequisites and notes.
2. In Superuser mode, type `configure terminal` and press <Enter>.
ACMEPACKET# **configure terminal**
3. Type `media-manager` and press <Enter> to access the signaling-related configurations.
ACMEPACKET(configure)# **media-manager**
4. 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.
5. 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.
6. Save and activate your configuration.

To set the SIP connect option for a SIP session agent configuration:

1. Check the [Configuring SIP Connect Support \(455\)](#) 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 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 optionally configure the number of seconds you want to keep a contact in the registration cache before it is deleted. This is the option:

- **contact-cache-linger=XX**—Number of seconds to wait before a contact is deleted from the cache (where XX is the number of seconds)

To enable SIP Registration overload protection on your 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 (+) (**contact-cache-linger=XX**) where XX is the number of seconds to keep a contact in the cache before deleting it.
ACMEPACKET(sip-config)# **options +contact-cache-linger=5**
If you type either of these options without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

5. Save and activate your configuration.

Session Replication for Recording

The Net-Net SBC's session replication for recording (SRR) feature allows it to forward signaling and media packets for calls associated with a specific realm to a destination Call Recording Server (CRS).

How It Works

You configure session replication for recording in the call-recording-server element under the session-router path. Specify primary and backup realms in order to link these realms and the network interface on which they reside to the CRS you are configuring. The Net-Net SBC considers a CRS valid only if network interfaces exist for each of the primary and backup networks. An invalid CRS entry will be ignored.

You must also specify addresses for media and signaling packets in your configuration. CRSs consist of both a signaling and media address, which are used to send the replicated packets. Each CRS object is configurable with both primary and secondary destinations. The secondary addresses are used as alternate destinations in case either of the primary CRS addresses is no longer reachable.

For a CRS to work, there MUST be either a primary media or primary signaling address configured. If both signaling and media packets are to be forwarded to the same address, you can leave the media address blank. However, if you provide only a media address, the Net-Net SBC forwards media and ignores signaling.

The destination addresses that you configure for media and signaling must be within the configured interface of their associated realm.

In a typical configuration, you only configure a CRS in the core realm of the network, however this is not always the case. If you configure a CRS in both the ingress and egress realms, the egress realm takes precedence.

You can enable High Availability by configuring a valid ping interval and method. When HA is configured, the Net-Net SBC creates an internal session agent with unique IP addresses for both primary and secondary CRSs. When signaling and media share the same address, the Net-Net SBC creates only one endpoint. You can have between 0-4 session agents depending on your configuration.

Any changes to a CRS are immediate; actively recorded sessions and newly initiated traffic are redirected to the newly configured destination. This is also the case when a CRS changes state in an HA node. When a primary CRS enters an inactive state, the Net-Net SBC replicates signaling and media traffic to the secondary CRS immediately, if active. Likewise, if the realm-config is configured to point to a new CRS object, the replicated packets are immediately sent to the new CRS. When a CRS is removed entirely from a configuration, recording is stopped for all active sessions and no new sessions are recorded.

The Net-Net SBC replicates all call recorded packets according to the IP-in-IP RFC 2003 encapsulation standard.

There are five different states in which a CRS can be:

- Unknown—Set when a CRS is first created. A CRS should never be in this state under normal operation.
- Not Monitoring—Set when the HA mode is disabled. When in this mode, the CRS table is populated with values configured for both the primary signaling and primary media address.

- Primary Active—Set when HA mode is enabled. Always takes precedence over secondary servers if all primary session agents are in an “In Service” state.
- Secondary Active—Set when HA mode is enabled. Takes place when one or all of the primary session agents are OOS and all of the secondary session agents are “In Service.”
- No Active—Set when HA mode is enabled. Takes place when one or more of both primary and secondary session agents are OOS.

Globally Unique Call ID for Call Replication

During IP call session replication recording (SRR), the Net-Net SBC records both media and signaling information and then sends them to a configured call recording server (CRS). It is the CRS’s responsibility to correlate signaling messages for specific calls, which can be difficult given that call information can traverse other network elements before reaching the CRS. The task of correlating the call information is simplified by the addition of a globally unique call ID.

For each SIP session, the Net-Net SBC can generate a unique call ID (UCID) that it inserts in SIP Request and Response messages for a call. The Net-Net SBC creates the UCID from a combination of the following put through an MD5 hash: transaction identifier, call identifier, plus the message’s branch and timestamp. Each UCID is truly unique and of a fixed length.

If it receives a dialogue-initiating request (or an initial-out-of-dialog request) with the UCID already present, the Net-Net SBC uses that UCID in its X-UCID header with the name “breadcrumbs.” It therefore becomes possible to trace the Net-Net SBCs traversed in that call flow if the message is recorded more than once. If breadcrumbs are present already, then the Net-Net SBC adds the current header to the end of the breadcrumbs list (as a comma-separated value) and replaces the UCID with a new one.

The ID appears as X-UCID in SIP messages, which you can see in the following examples.

UAC INVITE:

```

INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-11671-1-0
From: <sip:anonymous@anonymous.invali.d>;tag=1
To: <sip:service@192.168.1.61:5060>
Call-ID: 1-11671@192.168.1.60
CSeq: 1 INVITE
Max-Forwards: 70
Contact: <sip:sipp@192.168.1.60:5060>
X-UCID: jvo3lh1l65em8st286vdcn6sc3
Content-Type: application/sdp
Content-Length: 133
...

```

UAS INVITE:

```

INVITE sip:service@192.168.200.60:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK20q0pk30581g2eo6j141.1
From: <sip:anonymous@anonymous.invali.d>;tag=SDe9r2601-1
To: <sip:service@192.168.1.61:5060>
Call-ID: SDe9r2601-a205016a02b83cf347fefafa8c0c7437a6-06a3gu0

```

```
CSeq: 1 INVITE
Max-Forwards: 69
Contact: <sip:sipp@192.168.200.61:5090;transport=udp>
X-UCID: mh61vqhrukfsc9pg6sm863hj f7;breadcrumbs=jvo3lh1l65em8st286vdcn6sc3
P-Asserted-Identity: <sip:+3901251930010@ims.vodafone.it>
Content-Type: application/sdp
Content-Length: 140
...
```

UAS 180 Response:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-11671-1-0
From: <sip:anonymous@anonymous.invalid>;tag=1
To: <sip:service@192.168.1.61:5060>;tag=SDe9r2699-2
Call-ID: 1-11671@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:123456@192.168.1.61:5060;transport=udp>
Content-Length: 0
X-UCID: mh61vqhrukfsc9pg6sm863hj f7
```

X-UCID Notes

When you are using the globally unique call ID for SRR, remember that the Net-Net SBC:

- Does not insert the X-UCID in 100 Trying messages. Since the SIP session is established after the server side processes the SIP INVITE, a 100 Trying message will have already been sent.
- Inserts or modifies the UCID after any required SIP-NAT functions are performed.
- Treats a hairpinned call as though it had traversed two Net-Net SBCs. If the UCID is added to the first INVITE, that UCID becomes a breadcrumb parameter when the Net-Net SBC hairpins the call.
- Does not insert UCIDs in out-of-dialog requests such as REGISTER or OPTIONS messages. Typically, such messages follow a single response-request model and therefore are not part of the difficulty in correlating multiple messages to a single session.

License Information

You must have a valid license with the “Session Replication for Recording” option enabled to use this feature. Without this license option feature, you will not be able to configure the call-recording-server object under the session-router or the call-recording-server-id parameter under the realm-config.

CRS Capacity

For IP call session replication recording (SRR), the Net-Net SBC can support up to 256 call recording servers. No special configuration is required to use this number of CRSs.

ACLI Instructions and Examples

To configure the session replication for recording feature on the Net-Net SBC, use the **call-recording-server** configuration element.

To configure a CRS:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **call-recording-server** and press <Enter>.

```
ACMEPACKET(session-router)# call-recording-server
ACMEPACKET(call-recording-server)#
```
4. **name**—Enter the name you want to use for the CRS you are configuring.
5. **primary-realm**—Enter the primary realm to which you want this CRS to be associated. This must be an existing realm or the CRS will be considered invalid and this server will be ignored.
6. **primary-signaling-addr**—Enter the primary IP address you want to use as a destination for forwarding signaling packets.
7. **primary-media-addr**—Enter the primary IP address you want to use as a destination for forwarding media packets. If both the signaling and media primary addresses are the same, this parameter can be left blank.
8. **secondary-realm**—Enter the backup realm to associate with if the primary-network becomes unreachable. This must be an existing network interface or the CRS will be considered invalid and this server will be ignored.
9. **secondary-signaling-addr**—Enter the IP address you want to use as a destination for forwarding signaling packets if the address you entered in the **primary-signaling-addr** parameter becomes unreachable.
10. **secondary-media-addr**—Enter the IP address you want to use as a destination for forwarding media packets if the address you entered in the **primary-media-addr** parameter becomes unreachable.
11. **ping-method**—Enter the SIP method you want to be used for ping messages send to the CRS. This parameter only applies when the SIP protocol is implemented. If this parameter is left blank, HA is not used for this feature.
12. **ping-interval**—Enter the time in seconds to allow between the transmission of ping requests in an HA configuration. This parameter only applies when the SIP protocol is implemented. The default value is 0, meaning this parameter is disabled and HA is not configured for this feature. The valid range is:
 - Minimum—0 (disabled); 2 (minimum enabled)
 - Maximum—999999999
13. Save and activate your configuration.

To apply a CRS to a realm:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-config** and press <Enter>.

- ```
ACMEPACKET(configure)# media-config
ACMEPACKET(media-config)#
3. Type realm-config and press <Enter>.
ACMEPACKET(media-config)# realm-config
ACMEPACKET(realm-config)#
4. call-recording-server-id—Enter the name of the call recording server
associated with this realm.
```

**To include the UCID in SIP messages:**

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. **add-ucid-header**—Change this parameter from **disabled** (default) to **enabled** if you want the Net-Net SBC to include the UCID. You can use the UCID to correlate replicated SIP message information when you use SRR.
5. Save and activate your configuration.

## SIP Transport Selection

With this feature enabled, 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 UDP, or it might be behind a firewall. The UDP fallback option addresses this condition. If it is configured in SIP interfaces associated with an outgoing message and a TCP session cannot be established, the Net-Net SBC falls back to UDP and transmits the message. When the option is not present, the Net-Net SBC's default behavior is to return the SIP status message 513 Message too Large.

## ACL Instructions and Examples

You enable this feature per SIP interface by setting options that control the maximum UDP length and allow UDP fallback:

- **max-udp-length=X** (where X is the maximum length)—Sets the largest UDP packers that the 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.

You can set the global SIP configuration's **max-udp-length=X** option for global use in your SIP configuration, or you can override it on a per-interface basis by configuring this option in a SIP interface configuration.

- **udp-fallback**—When a request needs to be sent out on the SIP interface for which you have configured this option, the Net-Net SBC first tries to send it over TCP. If the SIP endpoint does not support TCP, however, then the Net-Net SBC falls back to UDP and tries the request again.

### To enable SIP Transport Selection:

1. In Superuser mode, type **configure terminal** and press <Enter>  
**ACMEPACKET# config terminal**
2. Type **session-router** and press <Enter> to access the **session-router** path.  
**ACMEPACKET(config)# session-router**
3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.  
**ACMEPACKET(session-router)# sip-interface**
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **max-udp-length=X** (where X is the maximum UDP length you want to set), and then press <Enter>.  
**ACMEPACKET(sip-interface)# options +max-udp-length=900**  
If you type **options max-udp-length=X**, you will overwrite any previously configured options. In order to append the new option to the **sip-interface**'s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **udp-fallback**, and then press <Enter>.  
**ACMEPACKET(sip-interface)# options +udp-fallback**  
If you type **options udp-fallback**, you will overwrite any previously configured options. In order to append the new option to the **sip-interface**'s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

## uaCSTA NAT Support

---

The Net-Net SBC offers User Agent Computer Supported Telecommunications Application (uaCSTA) support, which allows for the network address translation (NAT) of a key XML element in SIP INFO messages to use a phone's real contact URI.

### Overview

Certain customers who use a uaCSTA for third party call control have encountered difficulties with the XML in their SIP messages used to support business applications. In these cases, the XML—specifically the `<deviceID>` XML tag—carries encoded IP addresses that need to be changed as they traverse the Net-Net SBC.

The SIP business application allows users to click-to-dial another party using e-mail application clients. The user's click triggers the application server to send a uaCSTA SIP INFO message through the Net-Net SBC to the UA/phone. These SIP INFO messages contain XML with the user's Contact-URI. But the server is only aware of the Net-Net SBC's NAT'd Contact-URI and not the user's, so the XML in the SIP INFO is carrying incorrect information.

The XML element, then, needs to be NAT'd to the phone's real Contact-URI. This is especially important because of the broad use of SIP INFO messages, which instruct a phone to:

- Answer a call
- Hold a call

- Retrieve a call

All of these functions are available via a clickable interface on the e-mail application.

## How It Works

The Net-Net SBC performs the NAT to the <devicelD> XML tag only if it is configured to perform registration caching.

When the Net-Net SBC receives a SIP message from the core side and the request has:

- A Content-Type of application/csta+xml
- A Content-Length greater than 0

it parses the message's message body into an XML document. Should parsing fail, then the Net-Net SBC will forward the SIP INFO request without modification to the XML message body. Otherwise, the Net-Net SBC searches for the <devicelD> subelement within the XML document. If it finds the <devicelD> subelement, the Net-Net searches through its registration cache for a registered Contact that matches the value of the <devicelD>. If it finds a match, the Net-Net SBC replaces the value of the <devicelD> with that of the corresponding registered Contact. If the value of the <devicelD> is a Contact that the Net-Net SBC generates for a registered UA, the corresponding contact from the look-up would be the Contact of the registered UA.

These functions performed, the Net-Net SBC then reformats the SIP INFO request with the modified XML message body before sending it to the next hop. If there is no match found, then the Net-Net SBC forwards the SIP INFO request without modifying the XML message body.

## ACLI Instructions and Examples

Other than ensuring your Net-Net SBC is configured to perform registration caching, you do not need take any further steps.

## SIP Packet Cable Multi-Media

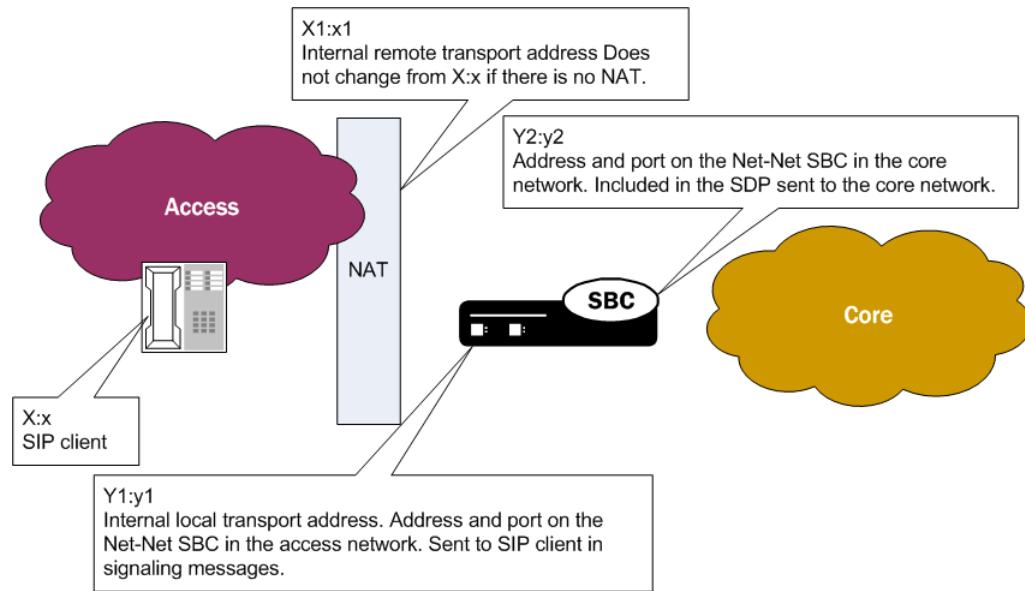
As a packet cable multi-media (PCMM) enhancement for SIP sessions key to next generation architectures, the Net-Net SBC can now include certain SDP attributes specifying media flow addresses in outgoing SIP messages. Previously, these address were hidden by the Net-Net SBC. Since SIP proxies and application servers in the core network, however, need to know these addresses to guarantee QoS for media flows in packet cable networks.

## How It Works

Certain options in the SIP interface configuration enable the Net-Net SBC to reveal address information on the core side.

When a SIP client in the access network sends and receives RTP media, the Net-Net SBC uses the SIP client's IP address and port (X:x) as its own internal remote transport address. The Net-Net SBC adds this information to outgoing SDP that it sends to the core side, and removes it from incoming SDP. If the SIP client sits behind a NAT, then the Net-Net uses the IP address and port produced from the NAT (X1:x1) process for insertion and removal. The SIP client sends RTP to an IP address and port (Y1:y1) on the Net-Net SBC, referred to as the internal local transport address; this information is included in SDP (included in SIP messages)

sent to the SIP client. Meanwhile, the Net-Net SBC also has an IP address and port (Y2:y2) in the core network. The far-end SIP UA sends RTP to this IP address and port, which are also included in SDP the Net-Net SBC sends to the core side.



To enforce QoS properly on the access side, the flow between the SIP client (or the SIP client's post-NAT IP address and port) and the internal local address must be revealed on the core side using SIP signaling messages.

## Details

To enable this enhancement, you set three parameters in the SIP interface configuration:

- **sdp-internals**—Establishes that local and remote transport addresses need to be added. This option must be enabled on the access-side SIP interface, which is where the Net-Net SBC receives SDP.
- **sdp-local=<name>**—Sets a name for the internal local transport port address that the Net-Net SBC inserts into outgoing SDP. This option is configured on the core-side SIP interface. This address is removed from incoming SDP from the core side to prevent attributes from being sent back to the core in a hairpinned call.
- **sdp-remote=<name>**—Sets a name for the internal remote transport address that the Net-Net SBC inserts into outgoing SDP. This option is also configured on the core-side SIP interface. This address is also removed from incoming SDP from the core side to prevent attributes from being sent back to the core in a hairpinned call.

Further, the Net-Net SBC determines whether or not to insert the SDP attributes based on a call's ingress and egress signaling realms:

| Address Information               | Calling-Side SDP                                                                                                                                                                                                                              | Called-Side SDP                                                                                                                                                                                                                               |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Internal local transport address  | Added to SDP when: <ul style="list-style-type: none"><li>• The ingress signaling realm's SIP interface has the sdp-internals option configured</li><li>• The egress signaling realm's SIP interface has a defined sdp-local option</li></ul>  | Added to SDP when: <ul style="list-style-type: none"><li>• The egress signaling realm's SIP interface has the sdp-internals option configured</li><li>• The ingress signaling realm's SIP interface has a defined sdp-local option</li></ul>  |
| Internal remote transport address | Added to SDP when: <ul style="list-style-type: none"><li>• The ingress signaling realm's SIP interface has the sdp-internals option configured</li><li>• The egress signaling realm's SIP interface has a defined sdp-remote option</li></ul> | Added to SDP when: <ul style="list-style-type: none"><li>• The egress signaling realm's SIP interface has the sdp-internals option configured</li><li>• The ingress signaling realm's SIP interface has a defined sdp-remote option</li></ul> |

## ACLI Instructions and Examples

In a typical configuration intended to send SDP to the core side with the inserted attributes, the access SIP interfaces have the **sdp-internals** option enabled, and the core SIP interfaces have the **sdp-local** and **sdp-remote** values configured.

### To set the access SIP interface for SDP insertion on the core side:

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-config**  
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. **options**—Set the options parameter by typing **options**, a <Space>, the option name **sdp-internals** with a “plus” sign in front of it, and then press <Enter>.  
ACMEPACKET(sip-interface)# **options +sdp-internals**

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

### To set the local and remote transport addresses for a core SIP interface:

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

2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.  

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

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

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

```
4. **options**—Set the options parameter by typing **options**, a <Space>, the option name preceded by a plus sign (+) (**sdp-local=<name>**, where the name is attribute name for the SDP), and then press <Enter>. Follow the same steps to add the **sdp-remote** option.  

```
ACMEPACKET(sip-interface)# options +sdp-local=Local_Turn
ACMEPACKET(sip-interface)# options +sdp-remote=PCMM_USERADD
```

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:

- INVITE | ACK | BYE | REGISTER | CANCEL | PRACK | OPTIONS | INFO | SUBSCRIBE | NOTIFY | REFER | UPDATE | MESSAGE | PUBLISH | “other” (unknown)

With this new tracking enhancement, the **show sipd** command has been updated with a new “method” argument which allows you to query statistics for a particular method for a given SIP agent, SIP interface, or SIP realm.

### ACLI Instructions and Examples

#### Enabling the SIP Method Tracking Enhancements

This section explains how to enable the expanded SIP method statistics tracking and how to view statistics for a particular SIP agent, interface, or realm method.

#### To enable or disable the expanded SIP Method statistics tracking:

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

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

```
ACMEPACKET(configure)# session-router
```
3. Type **sip-config** and press <Enter>.  

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

4. **extra-method-stats**—Enable this parameter if you want to use the expanded SIP Method tracking feature. The default is **disabled**. The valid values are:
  - enabled | disabled
5. Save and activate your configuration.

## 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.                                                                                                                          |

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 there is an rph-profile matching 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.

**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 <code>insert-r-value</code> parameter in the RPH policy applied to the NMC rule.                                                                                                                                                                                |
| Incoming request without RPH (matched by ETS DN) | If the <code>insert-r-value</code> 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 <code>override-r-value</code> for the policy is empty or if there is not RPH policy applied to the NMC rule.<br><br>If the <code>override-r-value</code> 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. fl ash` 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. fl ash`.

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, the incoming request has no RPH, or the call is H.323 and matches an NMC rule based on the ETS DN

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.

```
ACMEPACKET(rph-policy)# override-r-value ets.1
```

6. **insert-r-value**—Enter the value that the Net-Net SBC inserts into the RPH.

```
ACMEPACKET(rph-policy)# insert-r-value wps.1
```

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.

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)#[/pre]
    
```
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>.

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 default is accept. The valid values are:
 - **accept**—The call proceeds as it normally would
 - **reject**—The Net-Net SBC rejects the call with the 417 Unknown-Resource Priority status code
 - **priority**—The Net-Net SBC treats the call as a priority call

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)#[/pre]
    
```
3. Type **net-management-control** and press <Enter>.


```
ACMEPACKET(session-router)# net-management-control
ACMEPACKET(net-management-control)#[/pre]
    
```

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.

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**—Enable this parameter if you want to turn the NSEP feature on for this NMC rule. The default is **disabled**. The valid values are:
- enabled | disabled

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**—Enable this parameter if you want to turn the NSEP feature on for the global SIP configuration. The default is **disabled**. The valid values are:
- enabled | disabled

## **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. The default is **50**. The valid range is:
 - Minimum—0
 - Maximum—999999999
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.

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) #`
 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.

Setting Up NSEP for Session Agents

In earlier releases, the Net-Net SBC supports NSEP-related CAC 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 session exceed the limit, the Net-Net SBC rejects them with a 503 Service Unavailable message.

To configure NSEP limits for SIP session agents:

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

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

```
ACMEPACKET(config)# 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.

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 is signalled 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 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.

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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.

- ```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
3. Type session-agent and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI select command) the session agent that you want to edit.
4. reuse-connections—Enable or disable SIP TCP connection reuse. The default is none. This value disables the feature. The valid values are:
• tcp | none
5. tcp-reconn-interval—Enter the amount of time in seconds before retrying a TCP connection. The default for this parameter is 0. The valid range is:
• Minimum—0, 2
• Maximum—300
6. Save and activate your configuration.
```

## SIP TCP Keepalive

---

The Net-Net SBC supports a special TCP keepalive mechanism for SIP. By enabling this feature either for a session agent or for a SIP interface, you allow the Net-Net SBC to use standard keepalive probes to determine whether or not connectivity with a remote peer has been lost.

This feature adds to the Net-Net SBC's pre-existing TCP keepalive functionality that you can enable in the network parameters configuration. Using existing functionality, you can customize keepalive timing by:

- Specifying the number of unacknowledged packets the Net-Net SBC sends to the remote peer before it terminates the TCP connection.
- Specifying the number of seconds of idle time before TCP keepalive messages are sent to the remote peer.

You can now set three modes for TCP keepalive for session agents and SIP interfaces:

- **none**—(Default) Keepalives are not enabled for use with the session agent/SIP interface; when you select this setting for a session agent, it will use the setting for this feature from the SIP interface.
- **enabled**—Keepalives are enabled for the session agent/SIP interface.
- **disabled**—Keepalives are disabled for the session agent/SIP interface.

Note that the setting for this feature for a session agent takes precedence over that for a SIP interface. In addition, the session agent offers you a way to set the re-connection interval.

## ACLI Instructions and Examples

This section shows you how to enable this feature for session agents and SIP interfaces.

### SIP TCP Keepalive for Session Agents

#### To enable SIP TCP keepalive for session agents:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
 ACMEPACKET# **configure terminal**  
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.  
 ACMEPACKET(configure)# **session-router**  
 ACMEPACKET(session-router)#
3. Type **session-agent** and press <Enter>.  
 ACMEPACKET(session-router)# **session-agent**  
 ACMEPACKET(session-agent)#
 

If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the session agent that you want to edit.
4. **tcp-keepalive**—Enable or disable standard keepalive probes to determine whether or not connectivity with a remote peer is lost. The default value is **none**. The valid values are:
  - none | enabled | disabled
 ACMEPACKET(session-agent)# **tcp-keepalive enabled**
5. Save and activate your configuration.

### SIP TCP Keepalive for SIP Interfaces

#### To enable SIP TCP keepalive for SIP interfaces:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
 ACMEPACKET# **configure terminal**  
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.  
 ACMEPACKET(configure)# **session-router**  
 ACMEPACKET(session-router)#
3. Type **sip-interface** and press <Enter>.  
 ACMEPACKET(session-router)# **sip-interface**  
 ACMEPACKET(sip-interface)#
 

If you are adding support for this feature to a pre-existing SIP interface, then you must select (using the ACLI **select** command) the SIP interface that you want to edit.
4. **tcp-keepalive**—Enable or disable SIP TCP keepalive. The default value is **none**. The valid values are:
  - none | enabled | disabled
 ACMEPACKET(sip-interface)# **tcp-keepalive enabled**
5. Save and activate your configuration.

## SIP Enforcement Profile and Allowed Methods

For this feature, you use a configuration called an enforcement profile that allows you to configure sets of SIP methods that you want applied to: the global SIP configuration, a SIP interface, a realm, or a SIP session agent. The enforcement

profile is a named list of allowed methods that you configure and then reference from the configuration where you want those methods applied.

## ACLI Instructions and Examples

### Setting Up and Enforcement Profile

#### To set up an enforcement profile:

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

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

```
ACMEPACKET(config)# 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 EnfProfile
```
5. **allowed-methods**—Enter a list of SIP methods that you want to allow for this set. The default value is **none**. Valid values are:
  - INVITE | REGISTER | PRACK | OPTIONS | INFO | SUBSCRIBE | NOTIFY | REFER | UPDATE | MESSAGE | PUBLISH

To enter multiple methods for the list, type the parameter name followed by a space, then the names of all methods you want to include each separated by a only a comma and in capital letters.

```
ACMEPACKET(enforcement-profile)# allowed-methods
INVITE,REGISTER,PRACK
```
6. Save and activate your configuration.

### Applying an Enforcement Profile

You can apply an enforcement profile to: the global SIP configuration, a SIP interface, a realm, or SIP session agent. This section shows you how to do all four. Remember that if you are adding this functionality to a pre-existing configuration, you need to select the configuration you want to edit.

#### To apply an enforcement profile to the global SIP configuration:

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

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

```
ACMEPACKET(config)# 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.

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.

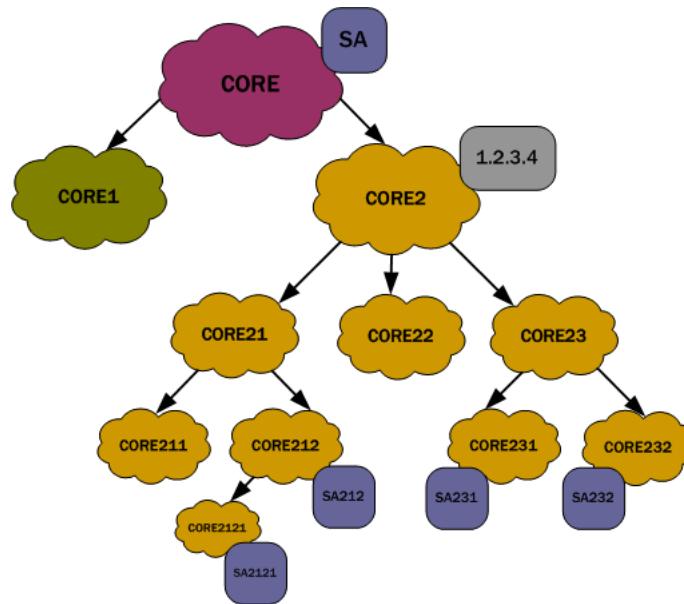
When it looks up a session agent with a realm, the Net-Net SBC first searches for an exact match for the IP address and realm combination. If this fails, it performs a second search if the desired realm has parents or children. The Net-Net SBC locates an entry in its repository of session agent information 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]
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]

Key	Session Agent (hostname[realm])
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 \(482\)](#) 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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.

- Unordered—Meaning that the endpoint can deliver data within regard for their stream sequence number

You set this preference in the network parameters configuration.

To set the SCTP delivery mode:

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

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

```

2. Type **system** and press <Enter>.

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

```

3. Type **network-parameters** and press <Enter>.

```
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#

```

4. **sctp-send-mode**—Leave this parameter set to its default (unordered) so data delivery can occur without regard to stream sequence numbering. If data delivery must follow stream sequence number, change this parameter to **ordered**.

5. Save and activate your configuration.

About Wildcarding

The 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 rrulefour
  allowed-methods disabled
  sdp-address-check
  subscribe-event
    event-type *
```

```

max-subscriptions           1
subscribe-event
  event-type                xyz
  max-subscriptions          0
last-modified-by            admin@console
last-modified-date          2008-11-11 12:49:27

```

In this example, the enforcement profile allows all subscriptions that are event type xyz for a user. But it allows only one maximum for every other subscription event type.

Monitoring

You can display the number of subscription dialogs per SUBSCRIBE event type using the **ACLI show registration sipd subscriptions-by-user** command. You can display this information per event type, or you can show data for all event types by wildcarding the event type argument.

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 \(486\)](#) 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 ACCLI **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
```
2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(configure)# 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
```
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.

```
ACMEPACKET(realm-config)# enforcement-profile
```
5. Save and activate your configuration.

STUN Server

The Net-Net SBC supports RFC 3489, which defines Simple Traversal User Datagram Protocol (UDP) through Network Address Translators (NATs). Known as STUN, this lightweight protocol that allows applications to:

- Discover the presence and types of both NATs and firewalls between themselves and the public Internet
- Determine the public IP addresses allocated to them by the NAT

SIP endpoints use the STUN protocol to find out the public IP addresses and ports for SIP signaling and RTP media transport. Then they can use the address and port information to create multimedia sessions with other endpoints on the public network.

You can define STUN servers functionality on a per-realm basis, allowing you set up multiple STUN servers.

About STUN Messaging

STUN messages uses six messages, three of which are used for Binding and three of which are used 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 fc3489bis.

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 REPONSE-ADDRESS attribute, if it was present in the Binding Request.
- The MAPPED-ADDRESS.

rfc3489bis Procedures

If the Binding Request contains the appropriate cookie in its Transaction ID, the server constructs a Binding Response populated with the XOR-MAPPED-ADDRESS attribute. That attribute will contain the source IP address and port from which the server saw the request come. Then the server encodes and sends the message to the client from the IP address and port on which the request was received. The message is sent to the IP address and port from which the request came.

Monitoring

- STUN Server Statistics—You can display statistics for the STUN server using the ACLI **show mbcd stun** command when the STUN server has been enabled. However, if the STUN server has not been enabled since the last system reboot, the command does not appear and no statistics will be displayed.
- STUN Protocol Tracing—You can enable STUN protocol tracing two ways: by configuration or on demand.
 - By configuration—The Net-Net SBC's STUN protocol trace file is called **stun.log**, which is classified as a call trace. This means that when the system configuration's call-trace parameter is set to enabled, you will obtain STUN protocol information for the system. As with other call protocol traces, tracing data is controlled by the log-filter in the system configuration.
 - On demand—Using the ACLI **notify mbcd log** or **notify mbcd debug** commands, you enable protocol tracing for STUN. Using **notify mbcd debug** sets the STUN log level to TRACE. You can turn off tracing using the **notify mbcd onlog** or **notify mbcd nodebug** commands. Using **notify mbcd nodebug** returns the STUN log level back to its configured setting.

ACLI Instructions and Examples

You configured STUN servers on a per-realm basis, one server per realm. To support that various NAT tests it describes, RFC 3489 requires that two different IP addresses and two different UDP port numbers be used for each server. So your STUN server will listen on a total of four STUN server ports. Although newer work does away with

this requirement, the 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# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(config)# 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(real-m-config)#
```
4. **stun-enable**—Set this parameter to **enabled** to turn STUN server support for this realm on. This parameter defaults to **disabled**, meaning STUN server support is off.
5. **stun-server-ip**—Enter the IP address for the primary STUN server port. The default for this parameter is 0.0.0.0.
6. **stun-server-port**—Enter the port to use with the **stun-server-ip** for primary STUN server port. The default is 3478.
7. **stun-changed-ip**—Enter the IP address for the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. This IP address must be different from than the one defined for the **stun-server-ip** parameter. The default for this parameter is 0.0.0.0.
8. **stun-changed-port**—Enter the port combination to define the CHANGED-ADDRESS attribute in Binding Requests received on the primary STUN server port. The default for this parameter is 3479.
9. Save and activate your configuration.

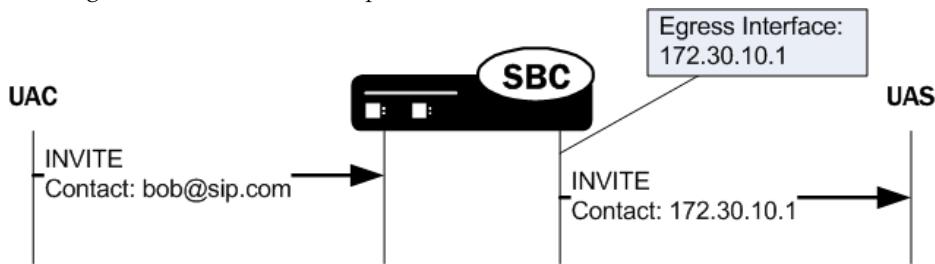
SIP GRUU

SIP Globally Routable User Agent (UA) URIs (GRUU) are designed to reliably route a SIP message to a specific device or end user. This contrasts with a SIP AoR which can refer to multiple UAs for a single user, thus contributing to routing confusion. The Net-Net SBC can perform different behaviors when it finds SIP GRUUs in Contact headers.

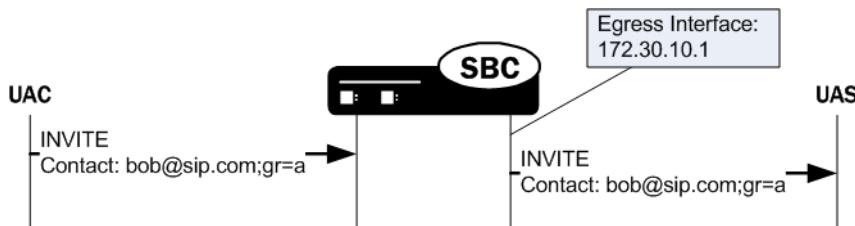
User agents supporting GRUU include a GRUU-identifying parameter in the Contact header of a dialog forming and target refresh requests. The Net-Net SBC scans for the GRUU parameter in the Contact header only when the endpoint it receives a request from is registered or when the pass-gruu-contact parameter 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 contact header's URI is replaced with the Net-Net SBC's egress interface. For example:

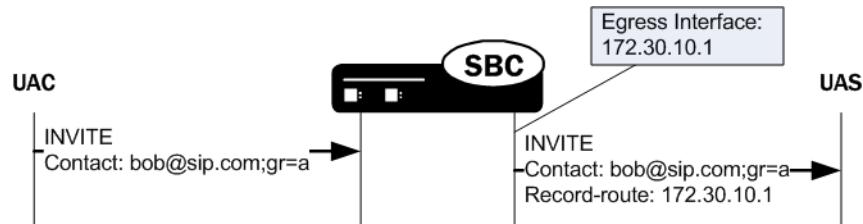


When the Net-Net SBC forwards a request where the original Contact header contains a GRUU, the contact header's URI is forwarded unchanged on the egress side of the call. For example:



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# configuration terminal
ACMEPACKET(configuration)#

```

2. Type `session-router` and press <Enter>.

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

```

3. Type `sip-config` and press <Enter>.

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

```

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

4. **`pass-gruu-contact`**—Set this parameter to enabled to parse for `gr` URI parameter in the contact header in non-registered endpoints' messages and then pass the messages thought the system.
5. **`options`**—Set the options parameter by typing `options`, a <Space>, the option name `scan-for-ms-gruu`. This option forces the Net-Net SBC to first scan for the `gruu` URI parameter, then the `gr` URI parameter.
6. Save and activate your configuration.

SIP ISUP Features

This section describes the Net-Net SBC's feaures for SIP ISUP.

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.

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**, 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# config terminal
ACMEPACKET(config)#

```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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**, or **redirection**. The **inherit** value is reserved for future use. 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.

To set up a SIP ISUP profile:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#

```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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.

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

```
ACMEPACKET(config)# 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, but 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 show 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.

To set up a SIP ISUP profile:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **confi gure terminal**
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.
 ACMEPACKET(configure)# **sessi on-router**
 ACMEPACKET(session-router)#
3. Type **sip-isup-profile** and press <Enter>.
 ACMEPACKET(session-router)# **si p-i sup-profi le**
 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 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.

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# **confi gure terminal**
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.
 ACMEPACKET(configure)# **sessi on-router**
 ACMEPACKET(session-router)#
3. Type **session-agent** and press <Enter>.
 ACMEPACKET(session-router)# **sessi on-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.

RTN 1605

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: `"($rul e1. $el em1. $0 + $rul e1. $el em2. @1) == $rul e2. $0"`. Equality operators can also be used with Boolean operators, as in this example: `"($rul e1. $0 == $rul e2. $1) & $rue3"`.

Equality operators always evaluate to either true or false.

Equality Operator Symbol	Short Description	Detailed Information
<code>==</code>	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.
<code>~=</code>	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.
<code>!=</code>	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.
<code>>=</code>	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.
<code><</code>	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.
<code>></code>	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
xxx_USER	The URI name of the header without any user parameters
xxx_PHONE	The URI user of the header as a phone number but without visual separators; may or may not contain a leading plus sign (+)
xxx_HOST	The URI host of the header
xxx_PORT	The URI port of the header; Value set to 5060 even if it is not actually in the message
CALL_ID	Resolves to the Call-ID of the current SIP message; added for convenience, and is a common store rule
TIMESTAMP_UTC	Timestamp is RFC 3339 format: 2009-10-10T22:00:09Z or YYY-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-C6.2.0, 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. Note that when you use the **delete** action on a multi-part MIME string that reduces a number of bodies down to one, the SIP message remains a multi-part MIME message with only one body part (and thereby avoids the header conflicting with the message itself).

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 `multi-part/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. Therefor, 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. For more information about binary body parts, refer to the **XX** section.

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:

| <b>parameter-format<br/>Setting</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| hex-ascii                           | Default.<br>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.<br><br>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<br>Setting | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bcd                         | <p>In ISUP speak, BCD refers to the binary forma 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.</p> <p>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.</p>                                                                                                                                                                                                    |
|                             | <p>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.</p>                                                                                                                                                                                                                            |
|                             | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| number-param                | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                             | <p>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.</p> |
|                             | <p>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 (+).</p>                                                                                                                                                                                                                                                                                                                                       |
|                             | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                        |

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

```
ACMEPACKET(config)# 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 **multi-part/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-parameter-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 \(502\)](#) 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 \(502\)](#) 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 mani p
 description
 header-rule
 name headerRule1
 header-name Date
 action add
 comparison-type case-sensitive
 msg-type reply
 methods
 match-value
 new-value
 element-rule
 name elemRule1
 parameter-name
 type header-value
 action add
 match-value-type any
 comparison-type case-sensitive
 match-value
 new-value "August 19, 1967"
 mime-rule
 name mimeRule1
 Content-Type application/SDP
 action manipulate
 comparison-type case-sensitive
 msg-type request
 methods
 match-value
 new-value
 mime-header
 name mimeHeaderRule1
 mime-header-name Content-Distribution
 action add
 comparison-type case-sensitive

```

```

 match-val ue
 new-val ue
 "si gnal ;
handl i ng=requi red"
 mi me-i sup-rule
 name
 content-type
 acti on
 comparis on-type
 msg-type
 methods
 match-val ue
 new-val ue
 i sup-spec
 i sup-msg-type
 mi me-header
 name
 mi me-header-name
 acti on
 comparis on-type
 match-val ue
 new-val ue
 "si gnal ;
handl i ng=opti onal "
 i sup-param-rule
 name
 parameter-type
 parameter-format
 hex, binary, ascii , bcd}
 acti on
 comparis on-type
 match-val ue
 new-val ue
 "si gnal ;
handl i ng=opti onal "
 i supRul e1
 # {0-256 speci fic type)
 {number-parameter,
 add
 case-sensi tive
 "si gnal ;

```

## Introduction

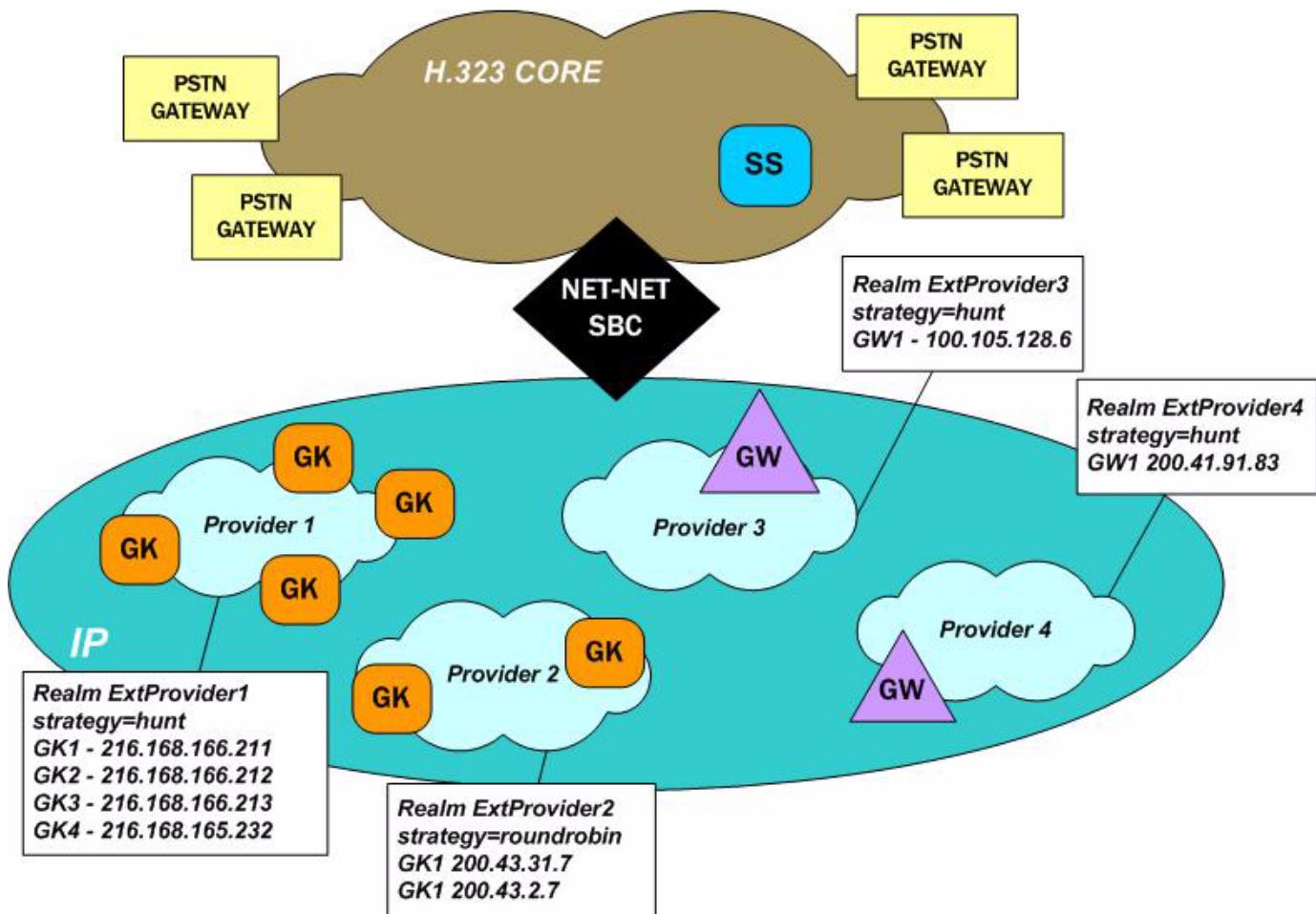
---

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 \(511\)](#), 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 [\(569\)](#).

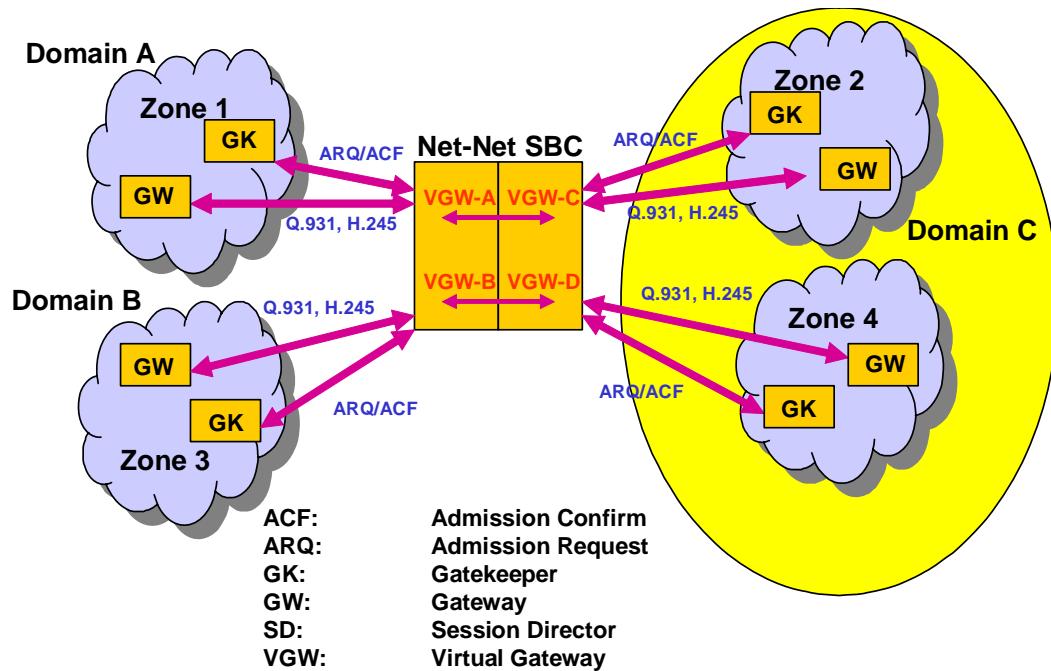
### Signaling Modes of Operation

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.

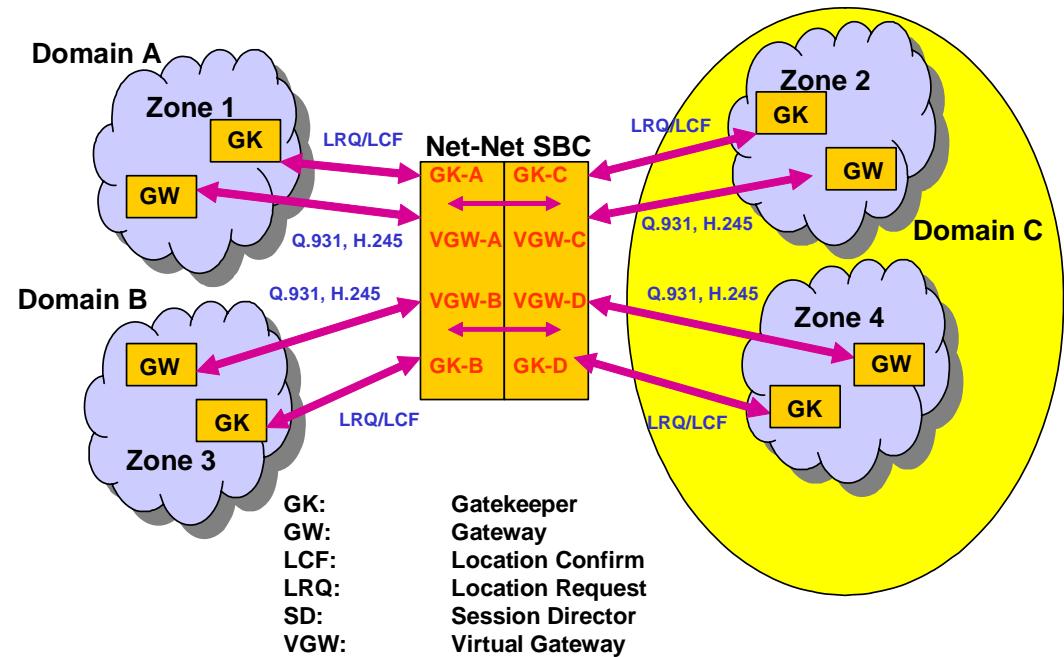
In the diagram above, the Net-Net SBC sends ARQs to the corresponding gatekeeper in its zone when a call is received on the associated interface. In this behavior, the Net-Net SBC acts as a gateway, complying with the H.323 standard, and registers with the configured gatekeeper in its assigned zone. 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.



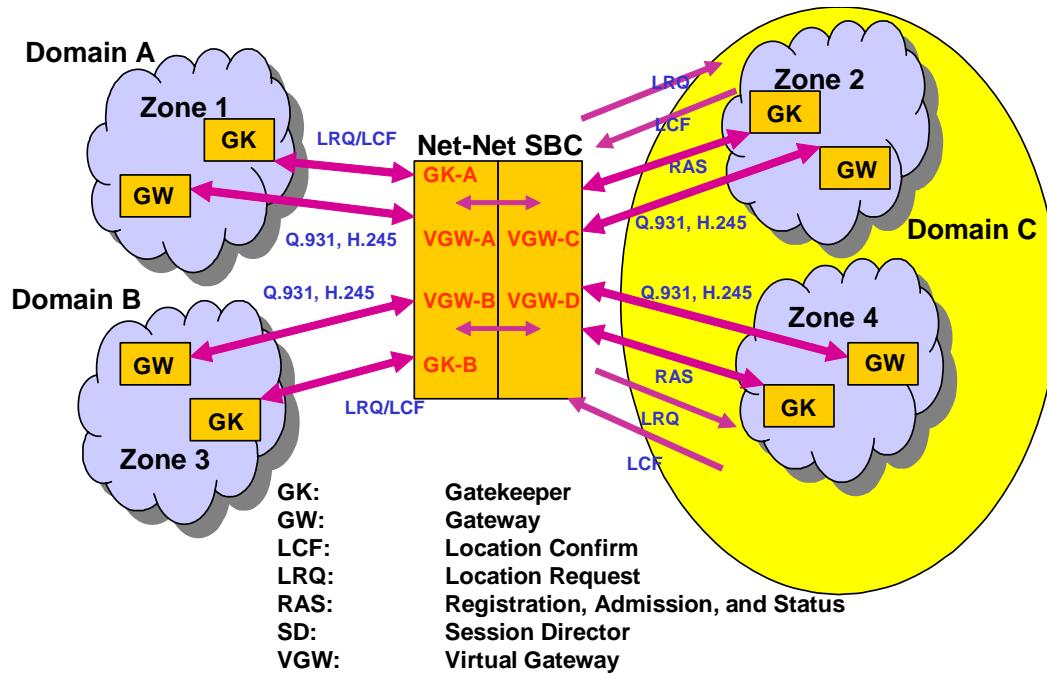
In this application, with the service mode (`isgateway`) parameter set to `disabled`, the Net-Net SBC responds to LRQs and issues LCFs and LRJs. It sends LRQs and LCFs/LRJs to the local IPv4 address for the H.323 interface. The Net-Net SBC responds to the LRQs by providing a signaling address that performs 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 \(743\)](#) 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 \(169\)](#)
- [Session Routing and Load Balancing \(743\)](#)

## 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:

|              |                                                             |
|--------------|-------------------------------------------------------------|
| state        | State of the H.323 protocol                                 |
| log-level    | Log level for H.323 stacks                                  |
| response-tmo | maximum waiting time in sec for response to a SETUP message |
| connect-tmo  | maximum waiting time in sec for establishment of a call     |
| options      | optional features/parameters                                |

### Accessing Global H.323 Parameters

**To access the global H.323 configuration parameters in the ACLI:**

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the session-related configurations.

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

3. Type **h323** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

ACMEPACKET(session-router)# **h323**

From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt. Access to the H.323 interface (**h323-stack**) configuration also appears.

## Global H.323 Settings

### To configure global H.323 parameters:

1. **state**—Enable or disable the state of H.323 signaling. The default value is **enabled**. Valid values are:
    - enabled | disabled
  2. **response-tmo**—Enter the amount of time in seconds that the Net-Net SBC waits between sending a Setup message and tearing it down if there is no response. The default value is **4** and we recommend you leave this parameter set to this value. The valid range is:
    - Minimum—0
    - Maximum—999999999

A response might be any of the following messages: Call Proceeding, Connect, or Alerting.
  3. **connect-tmo**—Enter the amount of time in seconds that the 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. The default is **32** and we recommend that you leave this parameter set to this value. The valid range is:
    - Minimum—0
    - Maximum—999999999

Receiving a Proceeding or Alert message from the endpoint does not keep this timer from expiring.
  4. **options**—Set any options for H.323 features that you want to use. This parameter has a global impact on H.323 behavior, rather than being applied on a per-interface basis. For more information about what parameters you want to configure, refer to the [International Peering with IWF and H.323 Calls \(565\)](#) section of this chapter.
- If you do not configure options for global H.323 behavior, none appears in the configuration display.
5. **log-level**—Set the process log level for monitoring all H.323 activity on the Net-Net SBC. The default is **INFO** and leaving this parameter set to this value provides an intermediate amount of detail in the logs. Other valid values are:

Note that any log level you set here overrides the log level you set in the system configuration's process log level parameter.

**Table 2: Log Levels**

| Numerical Code | Acme Packet Log Enumeration | Description                                                                     |
|----------------|-----------------------------|---------------------------------------------------------------------------------|
| 1              | EMERGENCY                   | Logs conditions of the utmost severity that require immediate attention.        |
| 2              | CRITICAL                    | Logs events of serious condition that require attention as soon as possible.    |
| 3              | MAJOR                       | Logs conditions indicating that functionality is seriously compromised.         |
| 4              | MINOR                       | Logs conditions indicating that functionality has been impaired in a minor way. |
| 5              | WARNING                     | Logs conditions indicating irregularities in performance.                       |
| 6              | NOTICE                      | For Acme Packet customer support.                                               |
| 7              | INFO                        |                                                                                 |
| 8              | TRACE                       |                                                                                 |
| 9              | 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# **config terminal**

2. Type **session-router** and press <Enter> to access the media-related configurations.

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

3. Type **h323** and press <Enter>.

ACMEPACKET(session-router)# **h323**

4. Type **h323-stacks** and press <Enter>.

ACMEPACKET(h323)# **h323-stacks**

ACMEPACKET(h323-stacks)#[/]

From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a ? at the system prompt. The display also includes H.323 service mode parameters.

## **Identity and State**

### **To set the identity and state of the H.323 interface:**

1. **name**—Enter a name for the H.323 interface using any combination of characters entered without spaces. For example: InternalGK1.
2. **state**—Enter the state of this H.323 interface. The default value is **enabled**. Valid values are:
  - enabled | disabled

## **Realm and Interface Associations**

### **To link this H.323 interface to a realm and to an outgoing H.323 interface:**

1. **realm-id**—Enter the identifier for the realm served by this H.323 interface. This parameter must be configured with a valid identifier value from a realm configuration.
2. **assoc-stack**—Enter the name of the outgoing H.323 interface that you want to associate with the H.323 interface you are configuring. To use realm bridging with static routing, you need to set the outgoing H.323 interface. If you do not enter a name, the Net-Net SBC uses dynamic, policy-based selection using the local policy.

## **H.323 Signaling Interface Settings**

You can set the following parameters to define basic settings for your H.323 interface. This is where you set the IPv4 address for opening sockets, the RAS and Q.931 ports, and the maximum number of Q.931 calls that you want to allow.

This is also where you establish Annex E alternate transport. Annex E supports multiplexed call signaling over UDP so that call volume and performance are potentially enhanced. If you do not configure Annex E support, then this H.323 interface does not listen for Annex E requests.

### **To configure H.323 interface settings:**

1. **local-ip**—Enter the IPv4 address that the H.323 interface uses when opening sockets; this is the default H.323 interface IPv4 address. You must use a valid IPv4 address. For example: 192.168.2.5. The default value is **0.0.0.0**.
2. **ras-port**—Enter the number of the port on the local IPv4 address (**local-ip**) on which the Net-Net SBC listens for RAS requests. We recommend that you set this parameter to its default, the well-known port **1719**. The valid range is:
  - Minimum—0

- Maximum—65535

If you set this parameter to **0**, the 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 its default, **1720**. The valid range is:
  - Minimum—0
  - Maximum—65535
4. **q931-max-calls**—Enter the maximum number of concurrent Q.931 calls you want to allow. The default value is **200**, however, this value should be less than the maximum number of calls you set when configuring [H.323 Features \(521\)](#). The valid range is:
  - Minimum—0
  - Maximum—65535

If the number of received Q.931 calls exceeds this number, the H.323 interface returns a *busy* state.

5. **alternate-transport**—Enter a list of one or more Annex E IPv4 address and port combinations for alternate transport. If you do not configure this list, then the 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**.

## **H. 323 System Resource Allocation**

You can set the following parameters to determine how many concurrent calls and concurrent channels you want to allow for each H.323 interface.

### **To allocate H.323 system resources:**

1. **max-calls**—Enter the maximum number of concurrent calls allowed on this H.323 interface. The default value is **200**. The valid range is:
  - Minimum—0
  - Maximum— $2^{32}-1$
2. **max-channels**—Enter the maximum number of concurrent channels allowed for each call associated with this H.323 interface. The default value is **6**. The valid range is:
  - Minimum—0
  - Maximum— $2^{32}-1$

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**—Enter the admission control of anonymous connections from an H.323 realm accepted and processed by this H.323 stack. The default value is `all`. The valid values are:
  - `all`—Allow all anonymous connections
  - `agents-only`—Allow requests from session agents only
  - `realm-prefix`—Allow session agents and addresses matching the realm prefix
2. **is-gateway**—To use this interface as an H.323 gateway, leave this parameter set to `enabled`, its default value. If you want to use this interface as an H.323 gatekeeper, set this parameter to `disabled`. Valid values are:
  - `enabled | disabled`
3. **terminal-alias**—Enter a list of one or more aliases that identify the H.323 interface. This value is either the gateway alias or the gatekeeper identifier, depending on the mode you configure for the interface. The aliases are set in the `sourceinfo` information element of outgoing ARQs.

### Configuring Gateway Only Settings

If you are using the H.323 interface as a gateway, you might want to set registration time-out and address prefix parameters.

### To configure gateway only settings:

1. **registration-ttl**—Enter the number of seconds before a registration becomes invalid. This value is used during the initial registration process. However, when a registration is confirmed, the time-to-live (TTL) value set by the gatekeeper in

the Registration Confirm (RCF) message overrides this value. The default value is **120**. The valid range is:

- Minimum—0
- Maximum— $2^{32}-1$

2. **prefixes**—Enter a list of prefixes for this H.323 interface. Possible prefix types include:

- H.323 ID | E.164 | URL | IPv4 address

These prefixes are sent from a gateway interface to a gatekeeper and indicate valid prefixes accepted by that interface for incoming calls. They are used if the interface is configured as a gateway (the **is-gateway** parameter is set to **enabled**).

Your entries for this parameter must appear as they do in the following example:

```
e164=17817566800 url=http://www.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**—To have the Net-Net SBC drop all RRQs, meaning that it does not acknowledge any requests, leave this parameter set to **disabled**, its default. To have the Net-Net SBC process any RRQs that arrive on this H.323 interface, set this parameter to **enabled**. Valid values are:

- enabled | disabled

When registration processing is enabled and the 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 the proxy mode that you want to use for the signaling only operation mode. For more information, refer to the [Signaling Only Operation \(545\)](#) description at the beginning of this chapter. Valid values are:

- H.225 | H.245

You can leave this field blank (default) if you are not using a proxy mode.

3. **h245-stage**—Set this field to the stage at which the 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. The default value is **connect**. Valid values are:

- Setup | Alerting | Connect | Proceeding | Early | Facility | noh245 | Dynamic

For more information, refer to the [Dynamic H.245 Stage Support \(541\)](#) 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:

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| <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    |

### 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 \(521\)](#) 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>. The system prompt changes to let you know that you can begin configuring individual parameters.   
ACMEPACKET(session-router)# **media-profile**  
From this point, you can configure media profiles parameters. To view all media profiles configuration parameters, enter a ? at the system prompt.
4. **name**—Enter the encoding name used in the SDP rtpmap attribute. You must enter a name to uniquely identify the media profile, and you will use this value to make lists of media profiles in H.323 interface configurations.
5. **media-type**—Leave this parameter set to its default, **audio**. Valid values are:
  - audio | video | application | data | image | text

6. **payload-type**—Enter the payload type number that corresponds to the encoding name you entered in Step 4. This value identifies the format in the SDP `m` lines. There is no default value for this parameter. To view possible values you might need, refer to this chapter's notes about [About Payload Types \(590\)](#), which includes a table of standard audio and visual encodings.

**Note:** When you use the RTP/AVP transport method, this value must be numeric.

7. **transport**—Enter the type of transport protocol used in the SDP `rtpmap` attribute. The default is **RTP/AVP**. Valid values are:
  - RTP/AVP | UDP
8. **req-bandwidth**—Enter the total bandwidth in kilobits that the media requires. The default value is **0**. The valid range is:
  - Minimum—0
  - Maximum— $2^{32}-1$
9. **frames-per-packet**—Enter the maximum number of frames to use per RTP packet. Leaving this parameters set to **0**, its default value means that it is not being used. The valid range is:
  - Minimum—0
  - Maximum—256

The interpretation of this value varies with codec type and with specific codec.

  - For frame-based codecs, the frame size is specific to each. For example, a G.729 frame contains ten milliseconds of audio, while a G.723.1 codec frame contains thirty milliseconds.
  - For sample-based codecs such as G.711, each frame contains one millisecond of audio.
10. **parameters**—Enter additional codec information. For example, the G.723.1 codec can have an additional `silenceSuppression` parameter.
11. **average-rate-limit**—Enter the maximum speed in bytes per second for the flow that this media profile applies to. The default value is **0**. The valid range is:
  - Minimum—0
  - Maximum—125000000
12. **peak-rate-limit**—Enter the peak rate for RTP flows in bytes per seconds. The default is **0**. The valid range is:
  - Minimum—0
  - Maximum—125000000
13. **max-burst-size**—Enter the maximum data size at peak rate in bytes. The default is **0**. The valid range is:
  - Minimum—0
  - Maximum—125000000
14. **sdp-bandwidth**—Enable this parameter to use the AS bandwidth modifier in the SDP in the conditions for the application specific bandwidth modifier. The default is **disabled**. Valid values are:
  - enabled | disabled

15. **sdp-rate-limit-headroom**—Specify the percentage of headroom to be added while using the AS bandwidth parameter while calculating the **average-rate-limit** (rate limit for the RTP flow). The default is 0. The valid range is:
  - Minimum—0
  - Maximum—100

## 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 ACCLI, 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> to access the session-related configurations.  
`ACMEPACKET(configure)# session-router`
3. Type **h323** and press <Enter>.  
`ACMEPACKET(session-router)# h323`
4. Type **h323-stacks** and press <Enter>.  
`ACMEPACKET(h323)# h323-stacks`  
`ACMEPACKET(h323-stacks)#{`  
 From this point, you can configure H.323 interface and service mode parameters. To view all H.323 interface parameters, enter a ? at the system prompt. The display also includes H.323 service mode parameters.
5. **fs-in-first-msg**—Enable this parameter if you want to include Fast Start fields in the first message that the Net-Net uses to respond to a Setup message. Usually, the first message sent is a Proceeding message. If you do not want Fast Start fields included, leave this parameter set to its default value **disabled**. Valid values are:
  - enabled | disabled
6. **call-start-fast**—Enable this parameter if you want Slow Start calls to be translated to Fast Start when this H.323 interface is chosen as the outgoing interface. If this parameter is **enabled**, **call-start-slow** has to remain disabled. The default value is **enabled**. Valid values are:
  - enabled | disabled
 If you set this parameter set to **disabled** (default), the outgoing call will be set up in the same mode as the incoming call.
7. **call-start-slow**—Enable this parameter if you want Fast Start calls to be translated to Slow Start when this H.323 interface is chosen as the outgoing

interface. If this parameter is **enabled**, **call-start-fast** has to remain disabled. The default value is **disabled**. Valid values are:

- enabled | disabled

If you leave this parameter set to **disabled**, the outgoing call will be set up in the same mode as the incoming call.

8. **media-profiles**—Enter the list of media profiles that you want to use when translating Slow Start calls to Fast Start. This information is used to open logical channels for the outgoing call.

If you enter the name of a media profile that does not exist, the Net-Net SBC will not perform translation. If you leave this parameter empty, the Net-Net SBC will not perform translation.

## RFC 2833: DTMF Interworking

This section explains the Net-Net SBC's support of transporting Dual Tone Multi-Frequency (DTMF) in Real-Time Transport Protocol (RTP) packets (as described in RFC 2833) to H.245 User Input Indication (UII).

Multimedia devices and applications must exchange user-input DTMF information end-to-end over IP networks. The Net-Net SBC provides the interworking capabilities required to interconnect networks that use different signaling protocols. Also, the Net-Net SBC provides DTMF translation to communicate DTMF across network boundaries.

The Net-Net SBC supports RFC 2833 to H.245 UII translation for H.323-to-H.323 calls, when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device that only uses H.245 UII.

### About RFC 2833

RFC 2833 specifies a way of encoding DTMF signaling in RTP streams. It does not encode the audio of the tone itself, instead a signal indicates the tone is being sent. RFC 2833 defines how to carry DTMF events in RTP packets. It defines a payload format for carrying DTMF digits used when a gateway detects DTMF on the incoming messages and sends the RTP payload instead of regular audio packets.

### About H.245 UII

H.245 provides a capability exchange functionality to allow the negotiation of capabilities and to identify a set of features common to both endpoints. The media and data flows are organized in logical channels. H.245 provides logical channel signaling to allow logical channel open/close and parameter exchange operations. The H.245 signaling protocol is reliable, which ensures that the DTMF tones will be delivered.

H.245 User Input Indication (UII) plays a key role in all the services that require user interaction. For video messaging, typical uses of UII include selection of user preferences, message recording and retrieval, and typical mailbox management functions. H.245 UII provides two levels of UII, alphanumeric and signal.

### About 2833 to H.245 UII Interworking

The Net-Net SBC provides 2833 to H.245-UII interworking by checking 2833-enabled RTP streams for packets matching the payload type number for 2833. It then sends the captured packet to the host for processing and translation to H.245 UII messages. A H.245 UII message received by the Net-Net SBC is translated to 2833 packets and inserted into the appropriate RTP stream.

## About DTMF Transfer

DTMF transfer is the communication of DTMF across network boundaries. It is widely used in applications such as interactive voice response (IVR) and calling card applications.

The multiple ways to convey DTMF information for packet-based communications include:

- In-band audio: DTMF digit waveforms are encoded the same as voice packets. This method is unreliable for compressed codecs such as G.729 and G.723
  - Out-of-band signaling events:
    - H.245 defines out-of-band signaling events (UII) for transmitting DTMF information. The H.245 signal or H.245 alphanumeric methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of through the RTP channel. The tones are transported in H.245 UII messages.
- All H.323 version 2 compliant systems are required to support the H.245 alphanumeric method, while support of the H.245 signal method is optional.
- RTP named telephony events (NTE): uses NTE to relay DTMF tones, which provides a standardized means of transporting DTMF tones in RTP packets according to section 3 of RFC 2833.

Of the three RTP payload formats available, the Net-Net SBC supports RTP NTE.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for 2833 telephone event packets using a SDP or H.245 Terminal Capability Set (TCS), which tells the other side what payload type number to use when sending the named event packets to the local device. Most devices use payload type number 101 for 2833 packets, although no default is specified in the standard.

The 2833 packet's RTP header also makes use of the timestamp field. Because events often last longer than the 2833 packets sending interval, the timestamp of the first 2833 packet an event represents the beginning reference time for subsequent 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

See RFC 2833 and draft-ietf-avt-rfc2833bis-07.txt for more information.

## Preferred and Transparent 2833

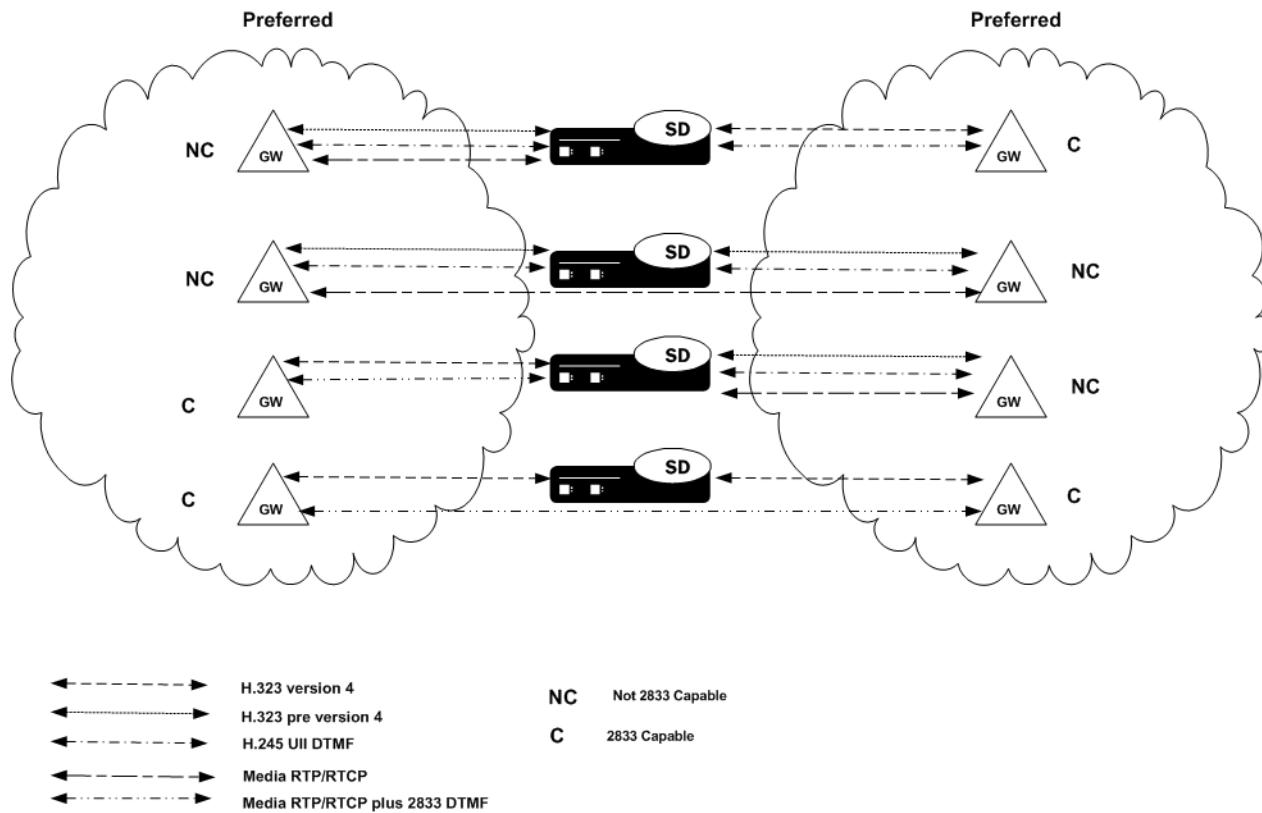
To support preferred (signaled) 2833 and transparent 2833, the Net-Net SBC provides 2833 detection and generation (if necessary) when the endpoint signals support for 2833.

- Preferred: the Net-Net SBC only generates and detects 2833 for endpoints if they negotiate support for 2833 through signaling
- Transparent: the Net-Net SBC behaves as it has prior to this release, offering and answering based on end-to-end signaling and transparently relaying 2833

## Preferred 2833 Support

If one side of the call, or a session agent, is configured for preferred 2833, the Net-Net SBC only generates and detects 2833 for endpoints if they signal support for 2833. The Net-Net SBC will offer 2833 in the TCS SDP, even if the originating caller did not.

- When the Net-Net SBC manages calls originating from a preferred source going to a preferred target, it:
  - Performs 2833 translation for an endpoint when the originating side requests 2833 but the target does not negotiate 2833
  - Allows 2833 to pass through if the originating side and target of the call are configured as preferred and negotiate 2833
- When the Net-Net SBC manages calls originating from a preferred source going to a transparent target, it:
  - Performs 2833 translation when the originating side requests 2833 but the target is configured as transparent and does not negotiate 2833.
  - Allows 2833 to pass through if the originating side and the target of the call are configured as transparent and negotiate 2833. The Net-Net SBC does not perform active translation because both ends support 2833.

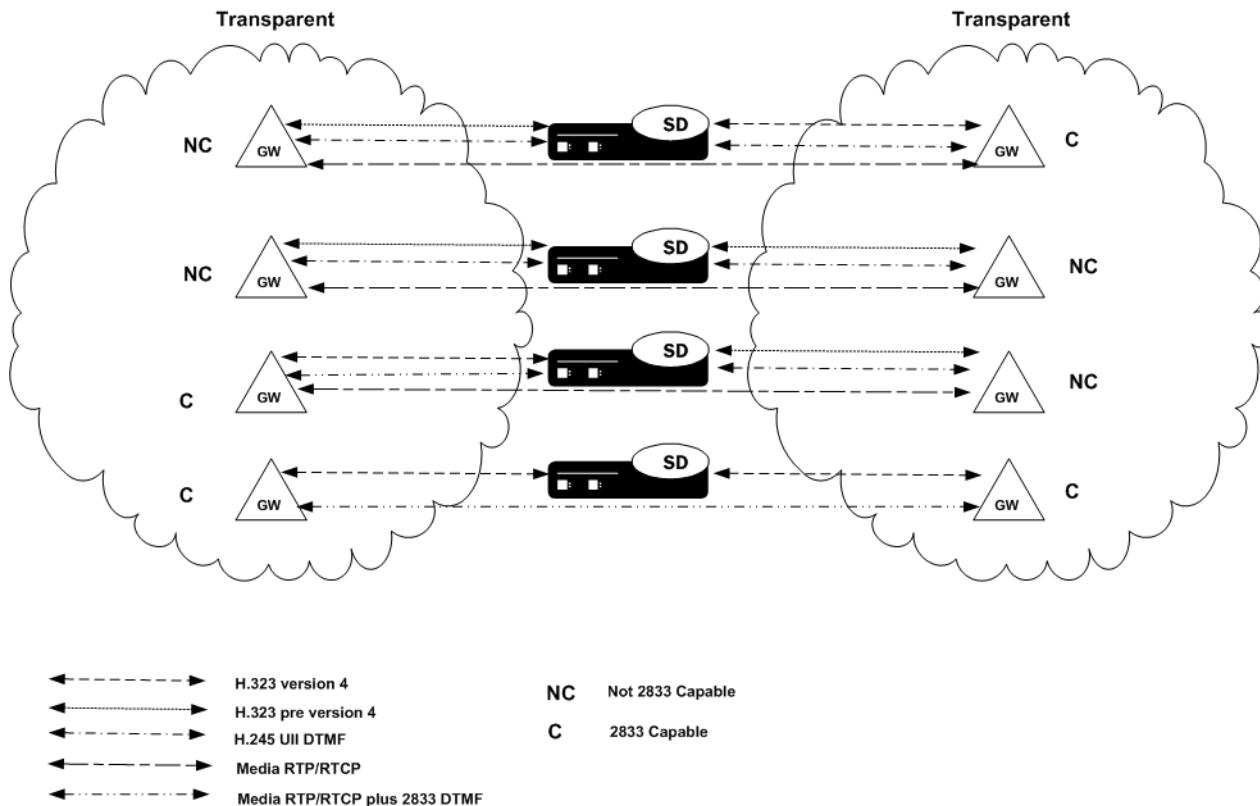


## Transparent 2833 Support

The default configuration of the Net-Net SBC for H.323 is transparent 2833. The Net-Net SBC passes on the offered capabilities to the next-hop signaling element. If the next-hop endpoint is for a transparent 2833 target, typical capability negotiation determines the DTMF method. The Net-Net SBC transparently relays the DTMF as it has in previous releases.

With transparent 2833, the Net-Net SBC acts as a typical B2BUA or B2BGW/GK. However when the target of the call is configured as preferred 2833, the Net-Net SBC:

- Relays the 2833 packets if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target signals 2833
- Performs 2833 translation if the originating endpoint does not signal 2833 and the next-hop endpoint for the preferred target does signal 2833
- Does not perform 2833 translation or transparently relay 2833 if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target (or even a transparent 2833 target) does not signal 2833.



### Basic RFC 2833 Negotiation Support

If H.323 or session agents on either side of the call are configured for preferred 2833 support, the Net-Net SBC supports end-to-end signaled negotiation of DTMF on a call-by-call basis. If the calling party is not configured for preferred support but sends 2833, the Net-Net SBC sends 2833 to the next-hop called party. If the calling party sends H.245 signals or alphanumeric UII, the Net-Net SBC sends H.245 signals or alphanumeric UII to the next-hop called party (if it is an H.323 next-hop).

The Net-Net SBC also supports hop-by-hop negotiation of DTMF capability on a call-by-call basis, if the signaling protocols or session agents on either side of the call are configured for preferred 2833 support.

## H.323 to H.323 Negotiation

The Net-Net SBC serves as the H.323 called gateway. It answers RFC 2833 audio telephony event capability in the version 4 H.323/H.245 TCS when it receives a call from an H.323 endpoint configured for preferred RFC 2833.

If the Net-Net SBC is the answering device, configured for preferred support, and the calling device sends 2833, the Net-Net SBC accepts the 2833 regardless of the next-hop's DTMF capabilities. The received dynamic RTP payload type is used for detecting 2833 packets, while the response dynamic payload type is used for generating 2833 packets.

The Net-Net SBC supports:

- RFC-2833 audio telephony events in the version 4 H.323/H.245 TCS as the H.323 calling gateway, when the Net-Net SBC calls an H.323 endpoint configured for preferred RFC 2833 support. The Net-Net SBC sends 2833 to the called party regardless of whether the calling party sends it.
- H.245 UII and RFC-2833 packets sent at the same time, to the same endpoint, even if only half of the call is being provided 2833 support by the Net-Net SBC.

If one half of the call supports H.245 UII, and the other half is being provided 2833 translation by the Net-Net SBC, the Net-Net SBC can also forward the H.245 UII it receives to the 2833 endpoint. For example, when the signaling goes through a gatekeeper or third party call control, sending the H.245 UII in the signaling path allows those devices to learn the DTMF digits pressed.

## Signal and Alpha Type Support

The Net-Net SBC supports:

- H.245 signal and alpha type UII in the H.323/H.245 TCS as the H.323 calling gateway when the:

- Net-Net SBC calls an H.323 endpoint configured for transparent 2833 support
- calling endpoint's target is configured as preferred

If the originating preferred side also sends 2833, the Net-Net SBC forwards it to the transparent side. The Net-Net SBC sends signal and alpha UII support to the called party regardless of whether the calling party sends it, if the call originates from a preferred side to a transparent side.

- H.245 alphanumeric UII for DTMF for H.323 endpoints that do not signal 2833 or contain explicit H.245 UII capability, for stacks configured for transparent 2833 support.

When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Net-Net SBC translates incoming H.245 UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Net-Net SBC relays the H.245 UII messages.

- H.245 signal type UII for DTMF for H.323 endpoints that do not signal 2833, but do signal explicit H.245 UII capability, for stacks configured for transparent 2833 support.

When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Net-Net SBC translates incoming H.245 signaled UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Net-Net SBC relays the H.245 UII messages if both sides support it.

## H.323 Endpoints

Because there are different H.323 endpoints based on different versions of H.323, the DTMF can be either be transferred out-of-band as UII or in-band using RFC 2833. Most H.323 endpoints:

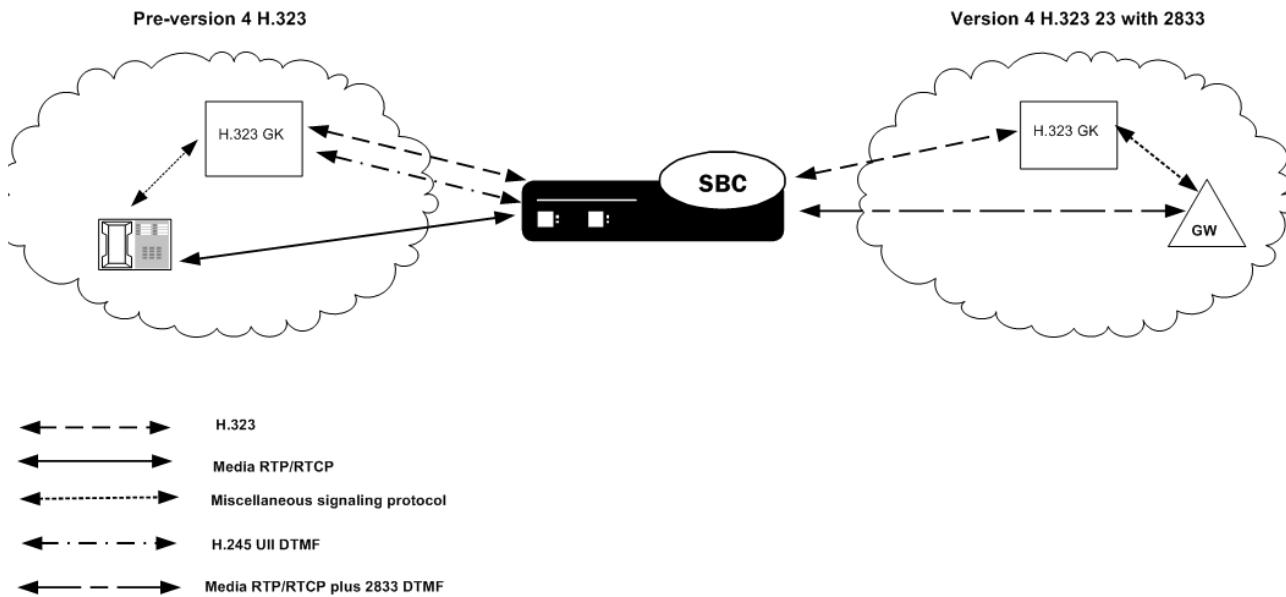
- version 4 and above support RFC 2833
- version 2 and pre-version 4 support UII-Signal
- version 1 and pre-version 2 support UII-Alphanumeric

## Translating H.245 UII to 2833 for H.323 Calls

A majority of H.323 endpoints are not version 4 H.323 compliant and do not support RFC 2833 for DTMF transfer. However, some networks include version 4 H.323 devices that require the DTMF events to be signaled in 2833 packets. Network-based version 4 H.323 gateways use RFC 2833 instead of H.245 UII. (Version 4 H.323 devices should support H.245 UII.)

The Net-Net SBC translates 2833 to H.245 UII for H.323-to-H.323 calls when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device which only uses H.245 UII.

The Net-Net SBC can translate H.245 UII to RFC2833 and back, based on the admin configuration and H.245 TCS exchanges. This translation enables DTMF to work end-to-end.



## ACLI Instructions and Examples

To configure RFC 2833 mode:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the media-related configurations.  
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.  
ACMEPACKET(session-router)# **h323**

4. Type **h323-stacks** and press <Enter>.  

```
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
From this point, you can configure H.323 stack parameters. To view all H.323 stack parameters, enter a ? at the system prompt.
```
5. **rfc2833-mode**—Set the RFC2833 mode. The default value is **transparent**. The valid values are:
  - **transparent**—The Net-Net SBC and H.323 stack behave exactly the same way as before and the 2833 or UII negotiation is transparent to the Net-Net SBC.
  - **preferred**—The H323 stack uses 2833 for DTMF transfer, which it signals in its TCS. However, the remote H323 endpoint makes the decision. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack reverts back to using UII. You configure the payload format by configuring the h323-config element.

To configure the RFC 2833 payload in preferred mode:

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

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

```
ACMEPACKET(configure)# session-router
```
3. Type **h323** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.  

```
ACMEPACKET(session-router)# h323
```

From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt.
4. **rfc2833-payload**—Enter a number that indicates the payload type the Net-Net SBC will use for RFC 2833 packets while interworking 2833 and UII. The default value is **101**. The valid range is:
  - Minimum—96
  - Maximum—127

You configure session agents with:

- payload type the Net-Net SBC wants to use for RFC 2833 packets while interworking 2833 and UII.  

The default value for this attribute is **0**. When this value is zero, the global rfc2833-payload configured in the h323-configuration element will be used instead. For SIP session agents, the payload defined in the SIP interface is used, if the SIP interface is configured with the preferred RFC 2833 mode.
- 2833 mode  

A value of transparent or preferred for the session agent's 2833 mode will override any configuration in the h323-stack configuration element.

To configure session agents:

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

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.  

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

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

```
4. **rfc2833-mode**—Set the RFC 2833 mode you want the session agent to use. The default is **none**. The valid values are:
  - **none**—2833 to UII interworking is based on the H.323 stack configuration.
  - **transparent**—The 2833 or UII negotiation is transparent to the Net-Net SBC. This overrides the H.323 stack configuration, even if the stack is configured for preferred mode.
  - **preferred**—2833 for DTMF transfer is preferred, which is signaled in the TCS. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack configured as preferred will revert back to using UII. This overrides any configuration in the h323-stack even if the stack is configured for transparent mode.
5. **rfc2833-payload**—Enter a number that indicates the payload type the session agent will use for RFC 2833 packets while interworking 2833 and UII. The default value is **0**. The valid range is:
  - Minimum—0, 96
  - Maximum—127

## H.323 Registration Proxy

The 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 \(592\)](#) 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 points, minus 1. Remember that you cannot overlap the Q.931 and dynamic H.245 port ranges. Notice that the higher the number of the start ports, the fewer ranges of ports you have remaining from which to choose.

**Table 3: 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# config terminal
```

2. Type **system** and press <Enter> to access the system-related configurations.

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

3. Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

4. Type **h323-stack** and press <Enter>.

```
ACMEPACKET(h323)# h323-stacks
```

```
ACMEPACKET(h323-stack)#{}
```

5. **q931-start-port**—Enter the number where you want the Q.931 port range to start. The default value is **0**. Valid values are:

- 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

6. **q931-number-ports**—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. The default value is **0**. Valid values are:

- 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

**Note:** If you have enabled process registration for this H.323 interface, this value must be set to zero because the interface is a gatekeeper that does not support the virtual call signaling address feature.

7. **dynamic-start-port**—Enter the number where you want the dynamic H.245 port range to start. The default value is **0**. Valid values are:

- 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

8. **dynamic-number-ports**—Enter the number of ports to be included in the Q.931 port range to use for the call signalling address forwarded in the RRQ. The default value is **0**. Valid values are:

- 0 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768

## H.323 Registration Caching

The Net-Net SBC can cache and proxy an H.225 RRQ between an H.323 endpoint and a gatekeeper. Registration caching has two benefits:

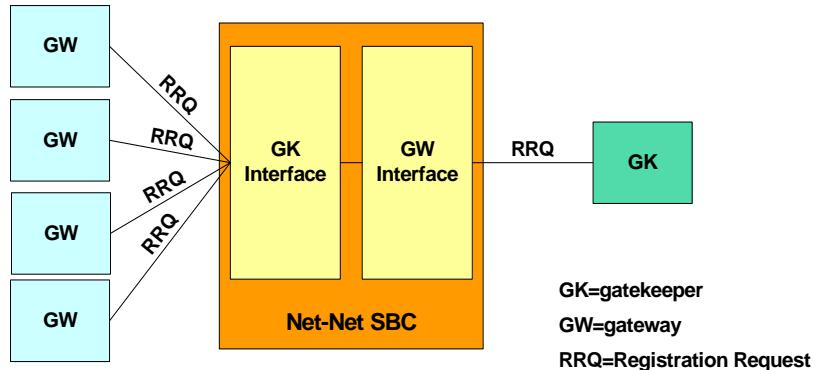
- It allows the aggregation of RRQs sent to a gatekeeper stack and proxies those requests through the gateway stack. If the external gatekeeper associated with the outbound (gateway) interface does not support additive registration, then the Net-Net SBC consolidates the requests by placing them all in 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 gkIdentifier 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-stacks
ACMEPACKET(h323-stack)#

```
5. **isgateway**—Enable H.323 stack functionality as a Gateway. Leave this parameter set to its default, `enabled`, so the H.323 stack runs as a Gateway. When this field is set to `disabled`, the H.323 stack runs as a Gatekeeper proxy. Leave this parameter for the service mode set to its default, `enabled`. Valid values are:
  - `enabled` | `disabled`
 Enabling this parameter ensures that registration with the gatekeeper upon startup. It also ensures that all calls will be preceded by an ARQ to the gatekeeper for admission control.
6. **registration-ttl**—Set the registration expiration parameter to the value of the `timeToLive` field in the RRQ sent to the gatekeeper. The default is `120`. The valid range is:
  - Minimum—`0`
  - maximum— $2^{32}-1$
 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. 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-stacks
ACMEPACKET(h323-stack)#

```
5. **`isgateway`**—Set this parameter to **disabled** to run the H.323 stack as a Gatekeeper proxy.
6. **`gatekeeper`**—Leave this parameter empty.
7. **`auto-discovery`**—Disable the Automatic Gatekeeper discovery feature upon start-up. Set this parameter to **disabled**.
8. **`gk-identifier`**—Set this parameter to the identification of the gatekeeper to which RRQs received on this interface must be proxied.
9. **`registration-ttl`**—Enter the number of seconds to set the `timeToLive` field in the RCF destined for an endpoint. If you do not configure another value, this timer is set to **120** seconds (default).

This value should not be set too high or too low:

- Setting a value that is too high causes the registration to be alive too long. If an endpoint reboots during this interval and re-registers with the same terminal aliases (but changes its call signaling address), the registration will be rejected with the reason `duplicateAliases`.
- 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 `duplicateAliases`. 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
 real-m-id private
 assoc-stack public
 local-ip 192.168.200.99
 max-calls 200
 max-channels 4
 registration-ttl 120
 terminal-aliases
 prefixes
 ras-port 1719
 auto-gk-discovery disabled
 multi-cast 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
 is-gateway enabled
 real-m-id public
 assoc-stack private
 local-ip 192.168.1.99
 max-calls 200
 max-channels 2
 registration-ttl 120
 terminal-aliases
 prefixes
 ras-port 1719
 auto-gk-discovery disabled
 multi-cast 0.0.0.0:0
 gatekeeper 192.168.1.50:1719
 gk-identifier gk-public.acme.com
 q931-port 1720

```

|                      |          |
|----------------------|----------|
| al ternate-transport |          |
| q931-max-calls       | 200      |
| h245-tunneling       | disabled |
| fs-in-first-msg      | disabled |
| call-start-fast      | disabled |
| call-start-slow      | disabled |
| media-profiles       | disabled |
| 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 allow 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.<br>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.<br>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(config)# **session-router**
3. Type **h323** and press <Enter>  
ACMEPACKET(session-router)# **h323**
4. Type **h323-stacks** and press <Enter>  
ACMEPACKET(h323)# **h323-stacks**  
ACMEPACKET(h323-stacks)#
  5. **h245-stage**—Set this field to the stage at which the 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. The default value is **Connect**. Valid values are:
    - Setup | Alerting | Connect | Proceeding | Early | Facility | noh245 | Dynamic

## H.323 HNT

This section explains how H.323 hosted NAT traversal (HNT) works and how to enable this capability on your 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.

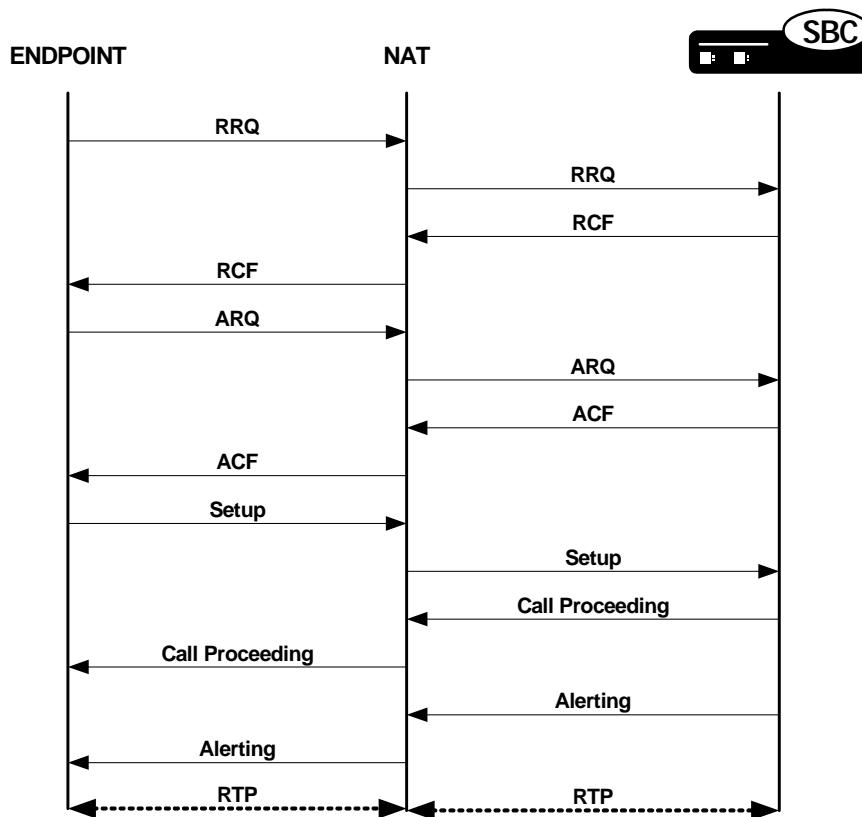
A second issue exists for media channels as the address specified in the H.323 OLC message will be behind the NAT and likewise unroutable. This is resolved by relying on the fact that the forward and reverse channels will utilize the same address and port on the endpoint. By sending media packets to 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# **config 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 you 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 **convertPNToE164** 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.

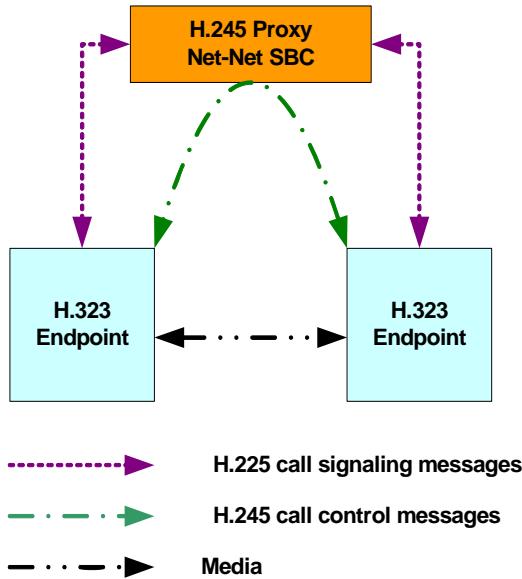
You have two options for the proxy mode:

- 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.

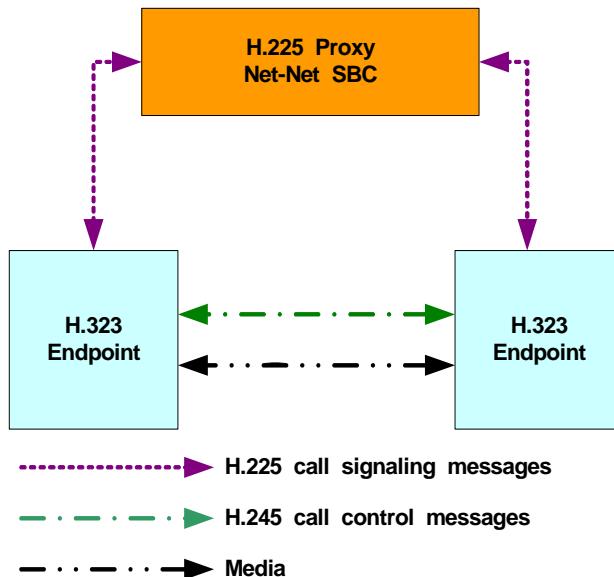
| Ingress                | Egress                 |
|------------------------|------------------------|
| Tunneled               | Tunneled               |
| Tunneled               | Separate H.245 session |
| Separate H.245 session | Tunneled               |
| Separate H.245 session | Separate H.245 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-stacks** and press <Enter>.  
 ACMEPACKET(h323)# **h323-stacks**  
 ACMEPACKET(h323-stacks)#
5. **options**—Set the options parameter to maintenanceProxy.

## Applying TCP Keepalive to the H.323 Interface

### To apply these settings individually per H.323 interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
 ACMEPACKET# **configure terminal**
2. Type **system** and press <Enter> to access the system-related configurations.  
 ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.  
 ACMEPACKET(session-router)# **h323**
4. Type **h323-stack** and press <Enter>.  
 ACMEPACKET(h323)# **h323-stacks**  
 ACMEPACKET(h323-stack)# **tcp-keepalive**
5. **tcp-keepalive**—Disable this parameter if you do not want the TCP keepalive network parameters to be applied. The default value is **disabled**. Valid values are:
  - enabled | disabled
6. Click OK at the bottom of the window to complete configuring TCP keepalives and the maintenance proxy.

## Automatic Gatekeeper Discovery

Available only when the H.323 interface is functioning as a gateway, this feature allows for automatic gatekeeper discovery on start-up.

This feature is based on the 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 a 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-stacks** and press <Enter>.  
 ACMEPACKET(h323)# **h323-stacks**  
 ACMEPACKET(h323-stacks)#
5. **auto-gk-discovery**—Enable this parameter to use automatic gatekeeper discovery. The default value is **disabled**. Valid values are:
  - enabled | disabled
6. **multicast**—Set this parameter to the address and port where the RAS Multicast Group listens. Your entries in this field will be comprised of an IPv4 address and port values separated by a colon. The default value is **0.0.0.0:0**.

## H.323 Alternate Routing

### Without Alternate Routing Enabled

You can configure your Net-Net SBC to try more possible routes within given time constraints and number of retries.

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 ACLI **connect-tmo** parameter, and this feature works only if there is more than one matching local policy next hop. If there is not more than one match, even if that match is a session agent group, then the call is only attempted once and the caller must retry it.

If the 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  |
| 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  |
| Tunneled 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# **config terminal**
2. Type **session-router** and press <Enter>.  
ACMEPACKET(config)# **session-router**
3. Type **h323** and press <Enter>.  
ACMEPACKET(session-router)# **h323**
4. **alternate-routing**—Enable or disable H.323 alternate routing. If you want to keep the pre-4.1 behavior where the Net-Net SBC only tries one matching local policy next hop, leave this parameter set to its default value **proxy**. Valid values are:
  - **recur | proxy**
5. **response-tmo**—Enter the time in seconds for the response time-out (or T303 timer). This is the amount of time allowed to elapse during which the 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**. The valid range is:
  - Minimum—0
  - Maximum—999999999
6. **connect-tmo**—Enter the time in seconds for the connect time-out (or T301 timer). This is the amount of time allowed to elapse during which the 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**.
  - Minimum—0
  - Maximum—999999999

## 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 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** x **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**—A session agent out of service until the no-response period expires (or until the Net-Net SBC receives a message from the session agent)
- **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:

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

```
ACMEPACKET# config terminal
```

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

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

- Type **h323** and press <Enter>.

```
ACMEPACKET(session-router)# h323
```

```
ACMEPACKET(h323)#{}
```

- 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:

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

```
ACMEPACKET# config terminal
```

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

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

- 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
```

- 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
```

- 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-Irj=3
```

If you type **options max-Irj=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(config)# session-router`
3. Type **h323** and press <Enter>.  
`ACMEPACKET(session-router)# h323`
4. Type **h323-stacks** and press <Enter>.  
`ACMEPACKET(h323)# h323-stacks`  
`ACMEPACKET(h323-stacks)#{`
5. Use the ACLI **select** command so can add this feature to an existing H.323 interface.  
`ACMEPACKET(h323-stacks)# select`
6. Set the **options** parameter by typing **options**, a <Space>, the option name **olcRejectTimer**, and then press <Enter>.  
`ACMEPACKET(h323-stacks)# options olcRejectTimer`
7. If you are adding this service to an H.323 interface that already exists, type **select** to select the interface to which you want to add the service. Then use the **options** command and prepend the option with a “plus” (+) sign.
  - If you know the same of the interface, you can type the name of the interface at the name: prompt and press <Enter>.

- If you do not know the name of the interface, press <Enter> at the name: prompt. A list of interfaces will appear. Type the number corresponding to the interface you want to modify, and press <Enter>.
- If you are adding service to an existing interface and type in the option without a “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +olcRejectTimer**.

## H.323 Per-Realm CAC

Building on the Net-Net SBC’s pre-existing call admission control methods, CAC can be performed based on how many minutes are being used by SIP or H.323 calls per-realm for a calendar month.

In the realm configuration, you can now set a value representing the maximum number of minutes to use for SIP and H.323 session using that realm. Although the value you configure is in minutes, the Net-Net SBC performs CAC based on this value to the second. When you use this feature for configurations with nested realms, the parent realm will have the total minutes for all its child realms (i.e., at least the sum of minutes configured for the child realms).

### How It Works

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 (H.323 Connect) and disconnect (H.323 Release Complete), regardless of whether media is released or not; there is no pause for calls being placed on hold.

If the number of minutes is exhausted, the 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.

**Note:** The Net-Net SBC does not reject GETS/NSEP calls based on monthly minutes CAC.

You can change the value for minutes-based CAC in a realm configuration at any time, though revising the value downward might cause limits to be reached. This value resets to zero (0) at the beginning of every month, and is checkpointed across both system in an HA node. Because this data changes so rapidly, however, the value will not persist across and HA node if both systems undergo simultaneous failure or reboot.

You can use the ACLI **show monthly minutes <realm-id>** command (where **<realm-id>** is the realm identifier of the specific realm for which you want data) to see how many minutes are configured for a realm, how many of those are still available, and how many calls have been rejected due to exceeding the limit.

### Caveats

Note that this feature is not supported for HA nodes running H.323.

## 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.

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# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.  

```
ACMEPACKET(config)# 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 and H.323 calls. By default, this parameter is set to zero (0), which disabled monthly minutes-based CAC. You can enter a value as high as 71582788.
6. Save and activate your configuration.

Use the ACLI show monthly-minutes command to see the following information:

- How many minutes are configured for a realm
- How many of those are still available
- How many calls have been rejected due to exceeding the limit

### To view information about SIP per user CAC using the IP address mode:

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
```

## 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 a an outgoing H.323 interface (stack) with **call-fast-start** set to enabled and there is a list of valid media-profiles in the configuration.

## H.323 BIS Enabled

There are certain cases in access deployments where the slow-start to fast-start conversion should not be applied. This is the case when the Setup message contains the Bearer Capability information element (IE), which signals BIS.

When you enable this feature and the 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>.  
ACMEPACKET# **config terminal**
2. Type **session-router** and press <Enter> to access the signaling-related configurations.  
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.  
ACMEPACKET(session-router)# **h323**
4. Type **options +bearerIndSetup** and press <Enter>.  
ACMEPACKET(h323-stacks)# **options +bearerIndSetup**

If you type **options bearerIndSetup** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

### To enable the H.323 BIS feature for a specific H.323 interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
ACMEPACKET# **config terminal**
2. Type **session-router** and press <Enter> to access the signaling-related configurations.  
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.  
ACMEPACKET(session-router)# **h323**
4. Type **h323-stacks** and press <Enter>.  
ACMEPACKET(h323)# **h323-stacks**  
ACMEPACKET(h323-stacks)#
  5. Select the H.323 stack to which you want to add H.323 BIS support.  
ACMEPACKET(h323-stacks)# **select**  
<name>:

For a list of configured H.323 interfaces (stacks), press <Enter> at the <name>: prompt. Then enter the number corresponding to the interface where you want to apply this feature.

6. Type **options +bearerIndSetup** and press <Enter>.

```
ACMEPACKET(h323-stacks)# options +bearerIndSetup
```

If you type **options bearerIndSetup** without the “plus” (+) sign, you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign as shown in the example above.

## **TOS Marking for H.323 Signaling**

You can configure your 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.

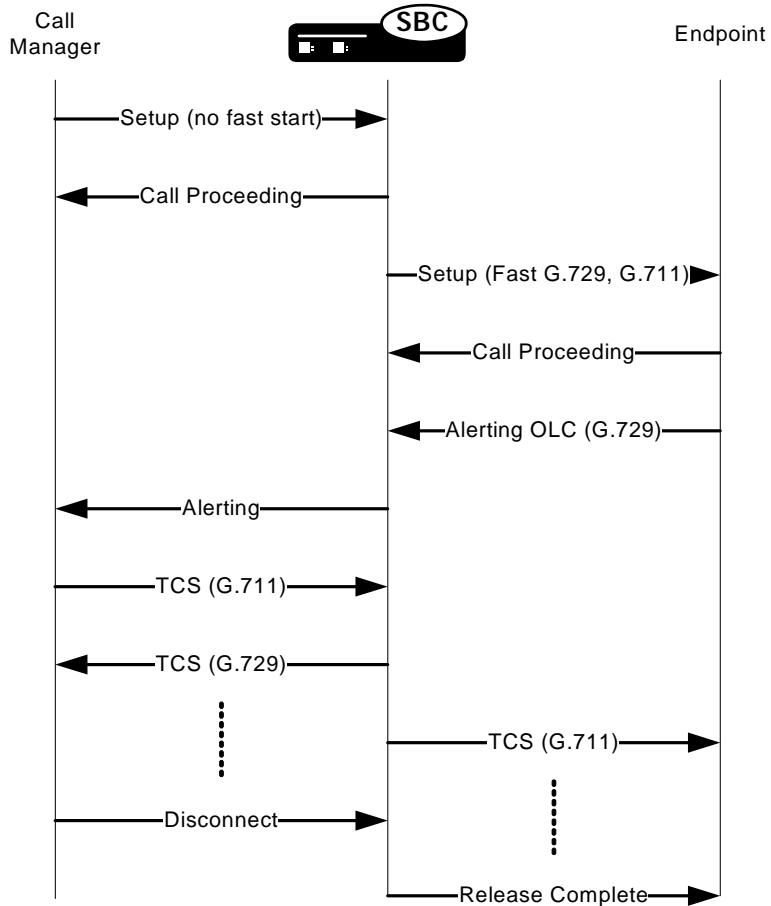
## **H.323 Codec Fallback**

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.

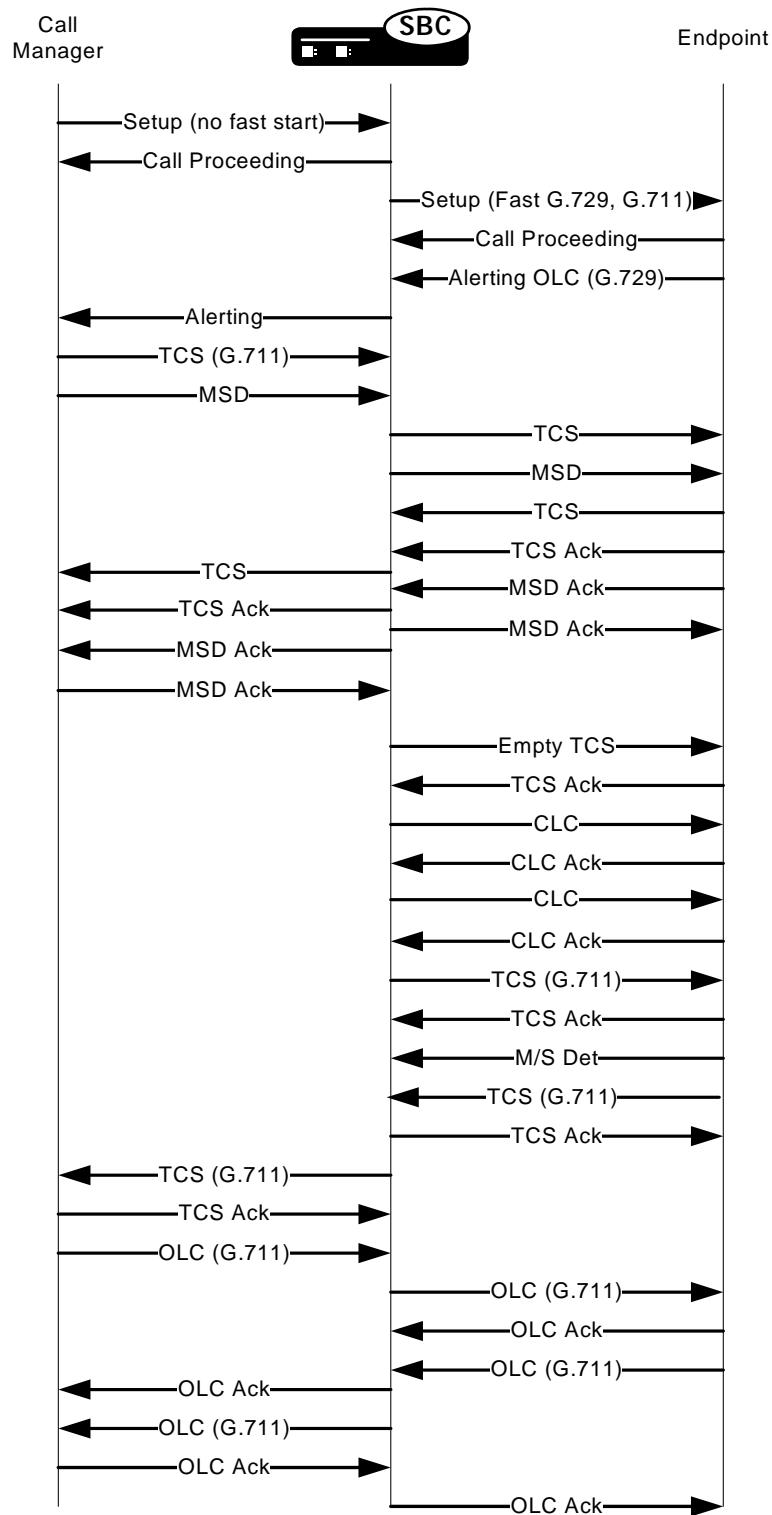


### Codec Fallback Enabled

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.

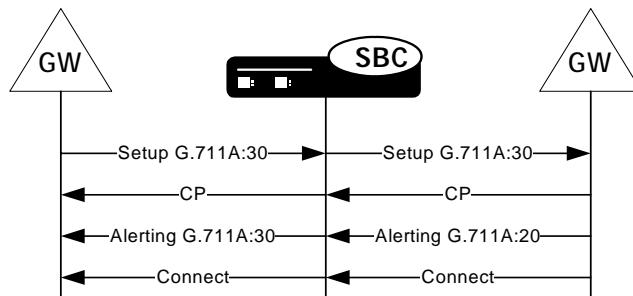
### To enable H.323 codec fallback:

1. In Superuser mode, type **configure terminal** and press <Enter>  
ACMEPACKET# **config terminal**
2. Type **session-router** and press <Enter> to access the signaling-related configurations.  
ACMEPACKET(config)# **session-router**
3. Type **h323** and press <Enter>. The system prompt will change to let you know that you can configure individual  
ACMEPACKET(session-router)# **h323**
4. **codec-fallback**—Enable or disable the H.323 codec fallback feature. The default value is **disabled**. Valid values are:
  - enabled | disabled

## H.323/TCS Media Sample Size Preservation

For H.323 fastStart calls, the 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.

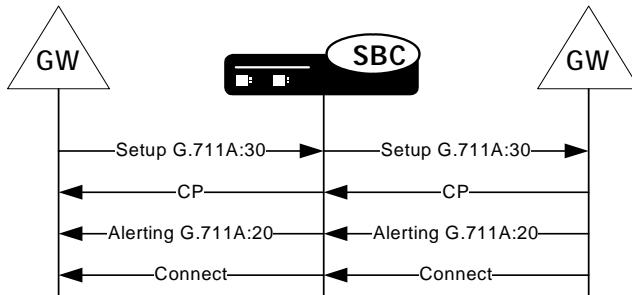
When this feature is disabled and in accordance with the ITU H.323 recommendation, the Net-Net SBC changes the packetization rate to the one used by the calling gateway if the one offered by the called gateway differs. In the following example, this means that the Net-Net SBC replaces the packetization 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-stacks** and press <Enter>.  
 ACMEPACKET(h323)# h323-stacks  
 ACMEPACKET(h323-stack)#  
 If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI **select** command) the one you want to edit.
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# **confi gure terminal**  
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.  
 ACMEPACKET(configure)# **sessi on-router**
3. Type **media-profile** and press <Enter>.  
 ACMEPACKET(session-router)# **medi a-profil e**  
 ACMEPACKET(medi a-profil e)#
4. **name**—Set the name of the generic video media profile to **generi cVi deo**. 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 **vi deo**.
6. **payload-type**—Set the payload type to use for the generic video media profile.
7. **transport**—Set the transport type to use for the generic video media profile.

8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic video media profile configuration:

```
medi a-profile
 name generi cVi deo
 medi a-type vi deo
 payl oad-type 99
 transport RTP/AVP
 req-bandwi dth 0
 frames-per-packet 0
 parameters
 average-rate-l i mi t 0
 sdp-rate-l i mi t-headroom 0
 sdp-bandwi dth di sabl ed
```

#### To set a media profile for generic audio support:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
 ACMEPACKET# **confi gure terminal**  
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.  
 ACMEPACKET(configure)# **sessi on-router**
3. Type **media-profile** and press <Enter>.  
 ACMEPACKET(session-router)# **medi a-profil e**  
 ACMEPACKET(medi a-profil e)#
4. **name**—Set the name of the generic audio media profile to **generi cAudi o**. 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 **audi o**.
6. **payload-type**—Enter the format in SDP m lines. No payload type number is assigned for newer, dynamic codecs. For RTP/AVP media-profile elements, this field should only be configured when there is a standard payload type number that corresponds to the encoding name. Otherwise, this field should be left blank. This field is used by the system to determine the encoding type when the SDP included with a session identifies the standard payload type on the em line, but does not include an a-rtpmap entry.
7. **transport**—Set the type of transport protocol to use for the generic audio media profile. The default value is **RTP/AVP**.
  - **UPD | RTP/AVP**
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic audio media profile configuration:

```
medi a-profile
 name generi cAudi o
 medi a-type audi o
 payl oad-type 104
 transport RTP/AVP
 req-bandwi dth 0
 frames-per-packet 0
```

```

parameters
average-rate-limit 0
sdp-rate-limit-headroom 0
sdp-bandwidth disabled

```

## International Peering with IWF and H.323 Calls

When you do not enable this feature, H.323 calls can default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option **cpnType=X**, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outgoing calls on the h323-stack.

The following is a list of possible **cpnType=X** option values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

## 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)#
5. Set the options parameter by typing **options**, a <Space>, the option name **cpnType=x** with a “plus” sign in front of it, and then press <Enter>.  
 ACMEPACKET(h323-stack)# **options +cpnType=x**  
 If you type **options** without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the h323-stack’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

## 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. <ul style="list-style-type: none"> <li>• Applies only to Fast Start OLC/OLC Ack messages embedded in H.225/Q.931 messages during call setup.</li> <li>• 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 OLCAck.</li> </ul> |
| 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.                                                                                                                                 |
| 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.                                                                                                                                                                                                                              |

## H.323 Interface Options

| Options       | Description                                                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                |

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 sourceInfo from the incoming Setup message to the outgoing Setup message. Otherwise, Net-Net SBC uses its own endpointType for the sourceInfo 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.                                                                                                                 |
| convertPNToE164     | Net-Net SBC converts the address type partyNumber to dialedDigits (E.164). Conversion applies to sourceAddress, destinationAddress, and destExtraCallInfo aliases in Setup messages.                                                                                                                                                                                                                            |

| Option                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| useCalledPNAsDestInfo | Net-Net SBC uses the H.225 called party number IE as the destinationInfo in ARQ and LRO requests. Since translation rules can be applied to the Called Party Number, the option enables digit normalization for RAS requests.<br>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: <ul style="list-style-type: none"> <li>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.</li> <li>Enabling the unique RRQ functionality (by setting the uniqueRRQRASAddress option) does not automatically enable H.323 RAS Port Mapping.</li> </ul> |
| srcCallSignallingPort | Enables use of the Q.931 port value for the port field in the sourceCallSignalAddress parameter in an H.225 Setup message. Useful for customers who configure a separate H.323 interface (stack) on the core side for each external IP-PBX.                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## H.323 Stack Monitoring

In releases prior to S-C6.2.0, the Net-Net SBC provides SNMP monitoring of H.323 session agents but not of the H.323 stacks themselves. The H.323 stack/interface configuration now provides a way for you to set alarm thresholds on a per-stack basis. When enabled, this alarm system ties into the max-calls value to send critical, major, or minor alarms when the number of calls approaches the threshold.

Each H.323 stack now has a threshold crossing alert (TCA) where you can set up three severity levels: critical, major, and minor. You can define one severity level or all three for each stack. To prevent the alarm from firing continuously as call volume through the stack varies, each severity level has a reset value below the TCA you set. In addition, each threshold value resets when:

- An alarm with a higher severity is triggered, or
- The built-in reset value for the threshold level is 1% less than the parameter value

RTN 1477

## ACLI Instructions and Examples

This section shows you how to configure H.323 stack monitoring for one H.323 stack configuration. This example shows one instance of the alarm-threshold sub-configuration being established; remember that you can set three—critical, major, and minor. Simply repeat the configuration steps to add more severity levels.

### To set up H.323 stack monitoring:

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

- ```

ACMEPACKET# config terminal
ACMEPACKET(config)#
2. Type session-router and press <Enter>.
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
3. Type h323 and press <Enter> to access the global H.323 configuration.
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
4. Type h323-stack and press <Enter>. If you are adding H.323 stack monitoring
to an existing H.323 stack configuration, then remember you must select the
stack you want to edit.
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
5. Type alarm-threshold and press <Enter> to configure this feature.
ACMEPACKET(h323-stack)# alarm-threshold
ACMEPACKET(alarm-threshold)#
6. severity—Enter the type of severity level for the alarm you want to define.
Choose from: critical, major, or minor. This value is required, and defaults to
minor.
7. value—Enter the percentage of the number of calls defined in the max-calls
parameter that triggers the alarm. For example, if you want to set a minor alarm
to fire when the call rate through the stack reaches half the max-calls value,
enter 50 (meaning 50%). The default value for this parameter is 0, which disables
the alarm.

Remember that if the number of calls falls to below 1% of the max-calls
threshold you set, the clear trap fires.
8. Save your work. You can see the data related to this feature using the ACLI
display-alarms and show h323 stack stack-alarms commands.

```

H.323 Automatic Features

This section describes H.323 features that are automatically enabled on your 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.

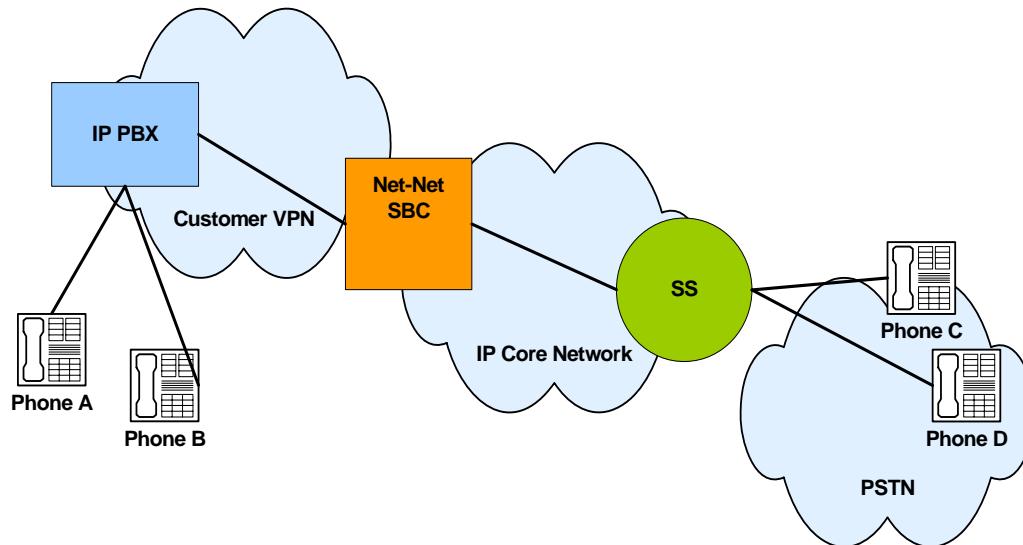
When sending an ARQ or an LRQ message to a gatekeeper, the Net-Net SBC sets the canMapAI field in that message to true. This setting indicates that the Net-Net SBC accepts modified destination information from the gatekeeper. If the resulting ACF or LCF contains destinationInfo and/or destExtraCallInfo 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

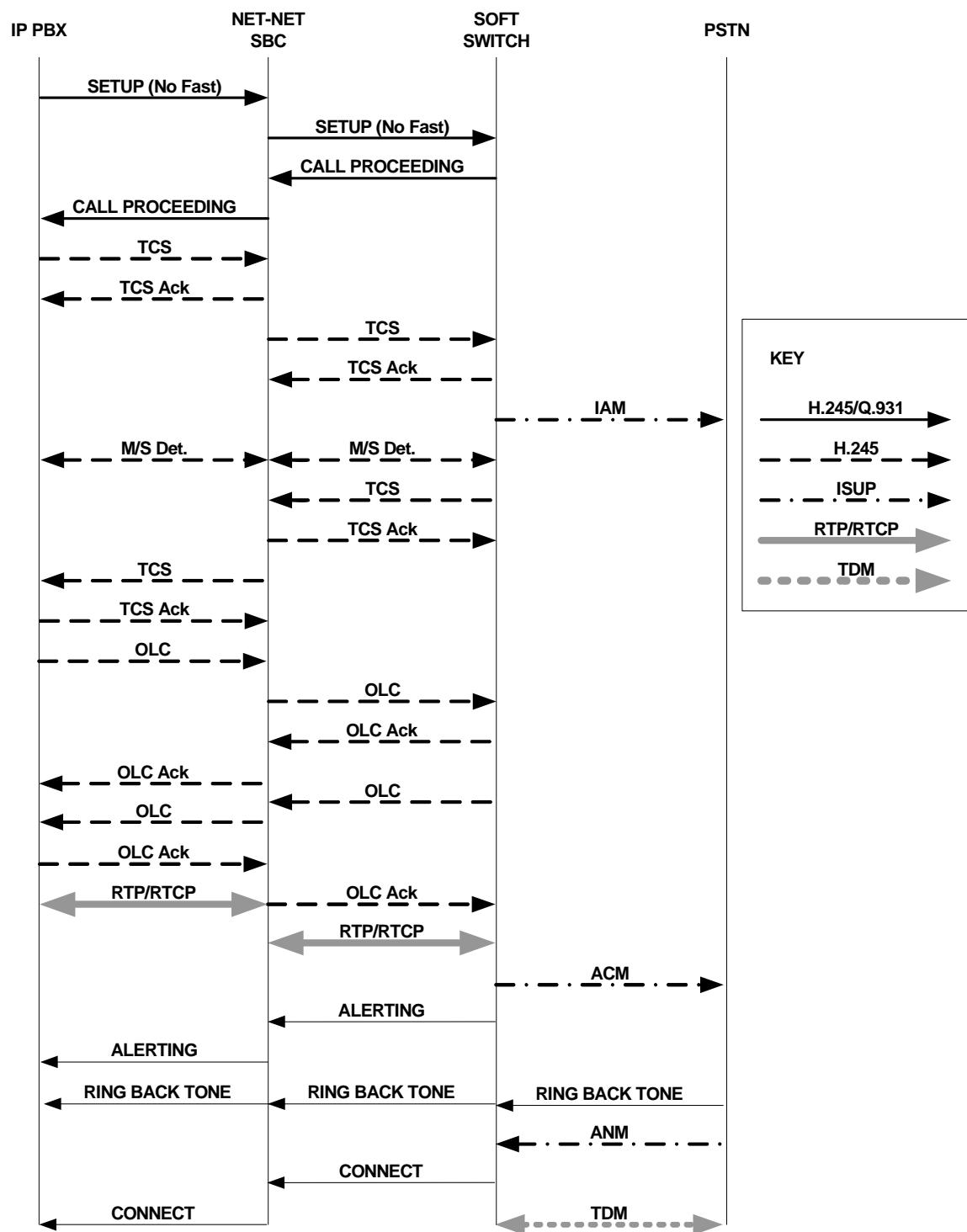
The following diagram show the signaling and media flows between the IP PBX and a softswitch. Note how the Net-Net SBC is position to mediate flows between the two devices.

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:

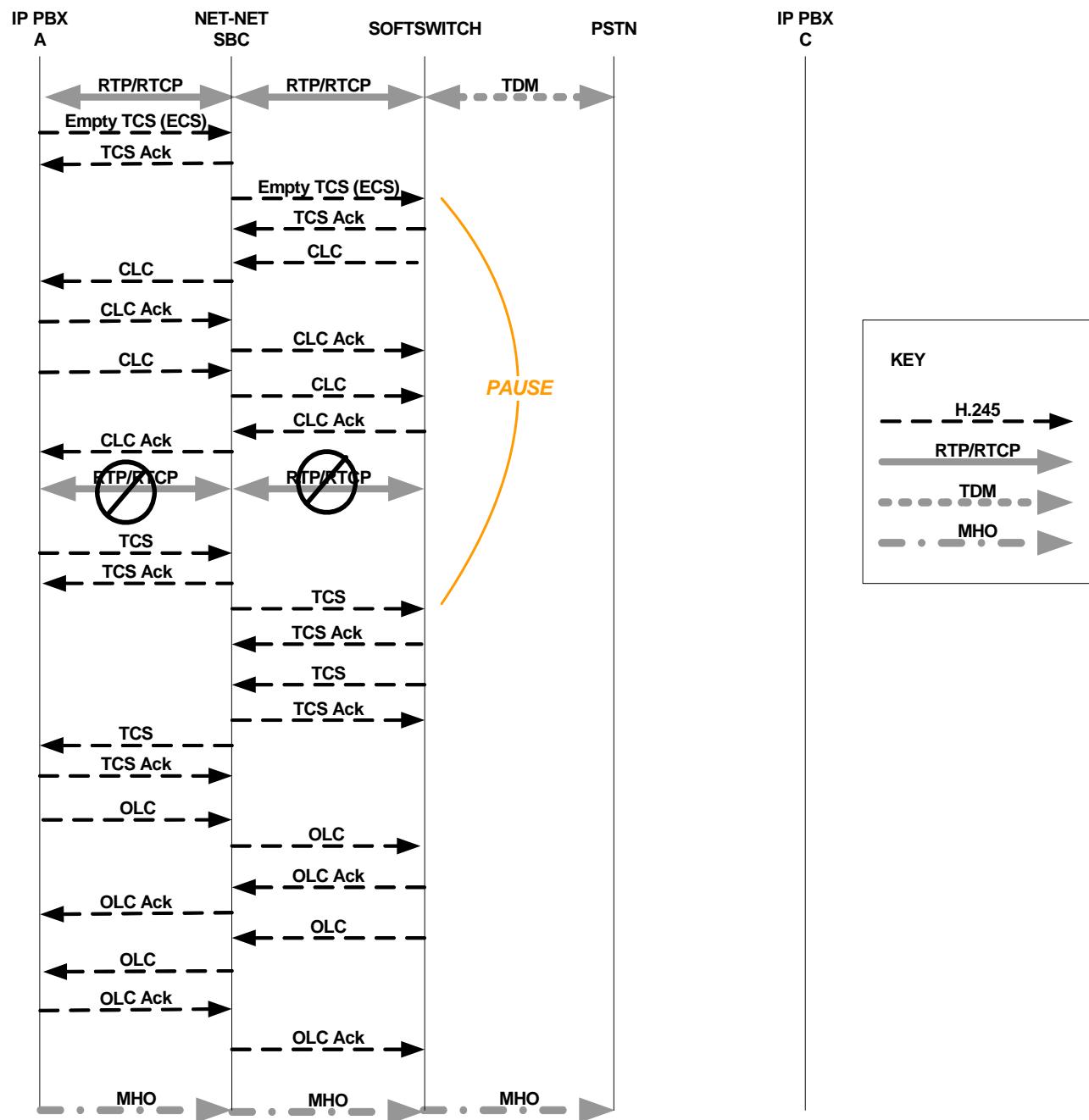
- Initial Address Message (IAM)
- Address Complete Message (ACM)
- Answer Message (ANM)



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.

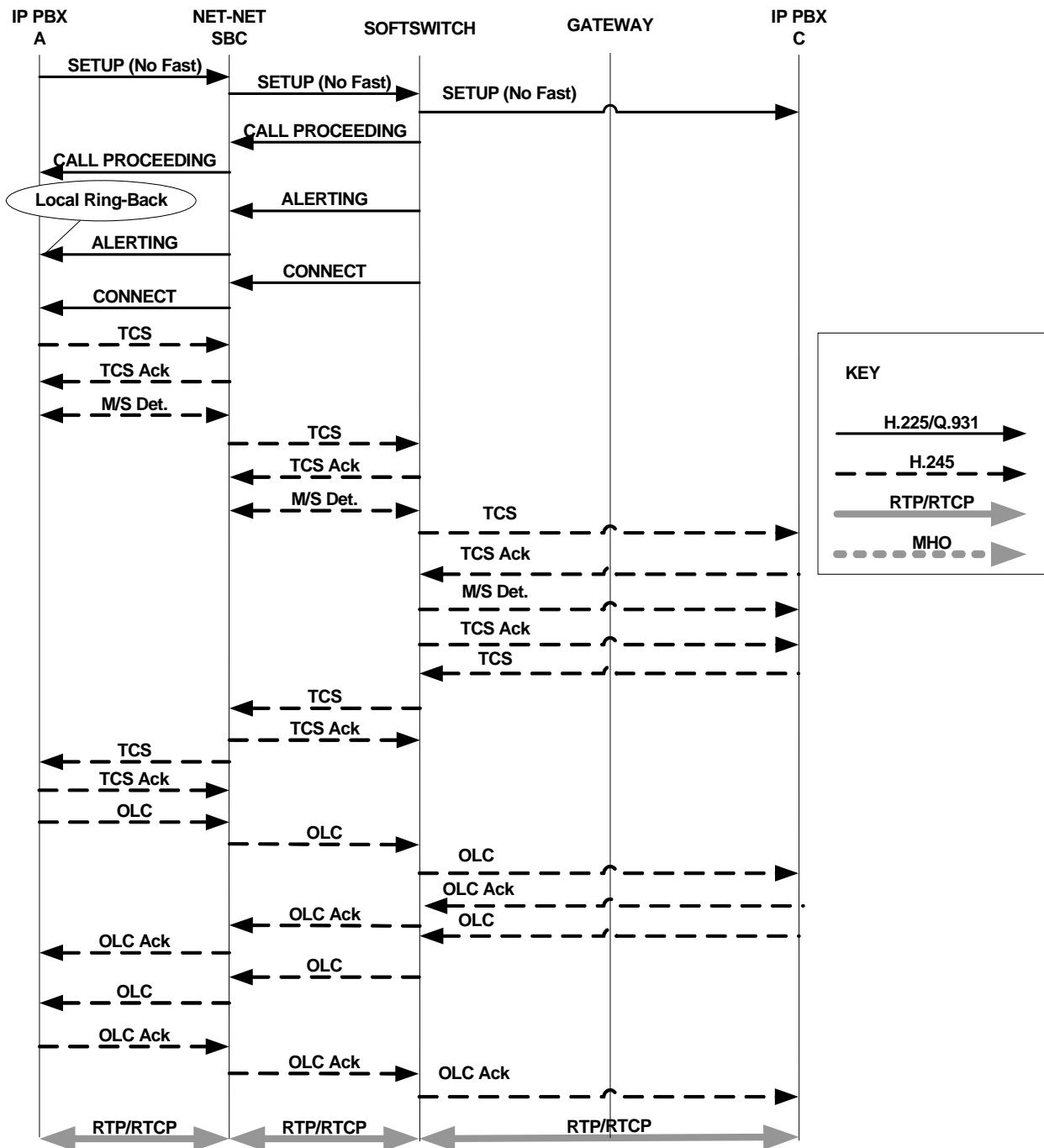
You can see in the call flow where the channels for transporting media are closed, and where the RTP/RTCP is stopped. This creates a pause for the call. With the Net-Net SBC mediating the process, IP PBX A and the softswitch exchange TCS and OLC messages that allow music on hold (MHO) to flow between IP PBX A and the gateway.

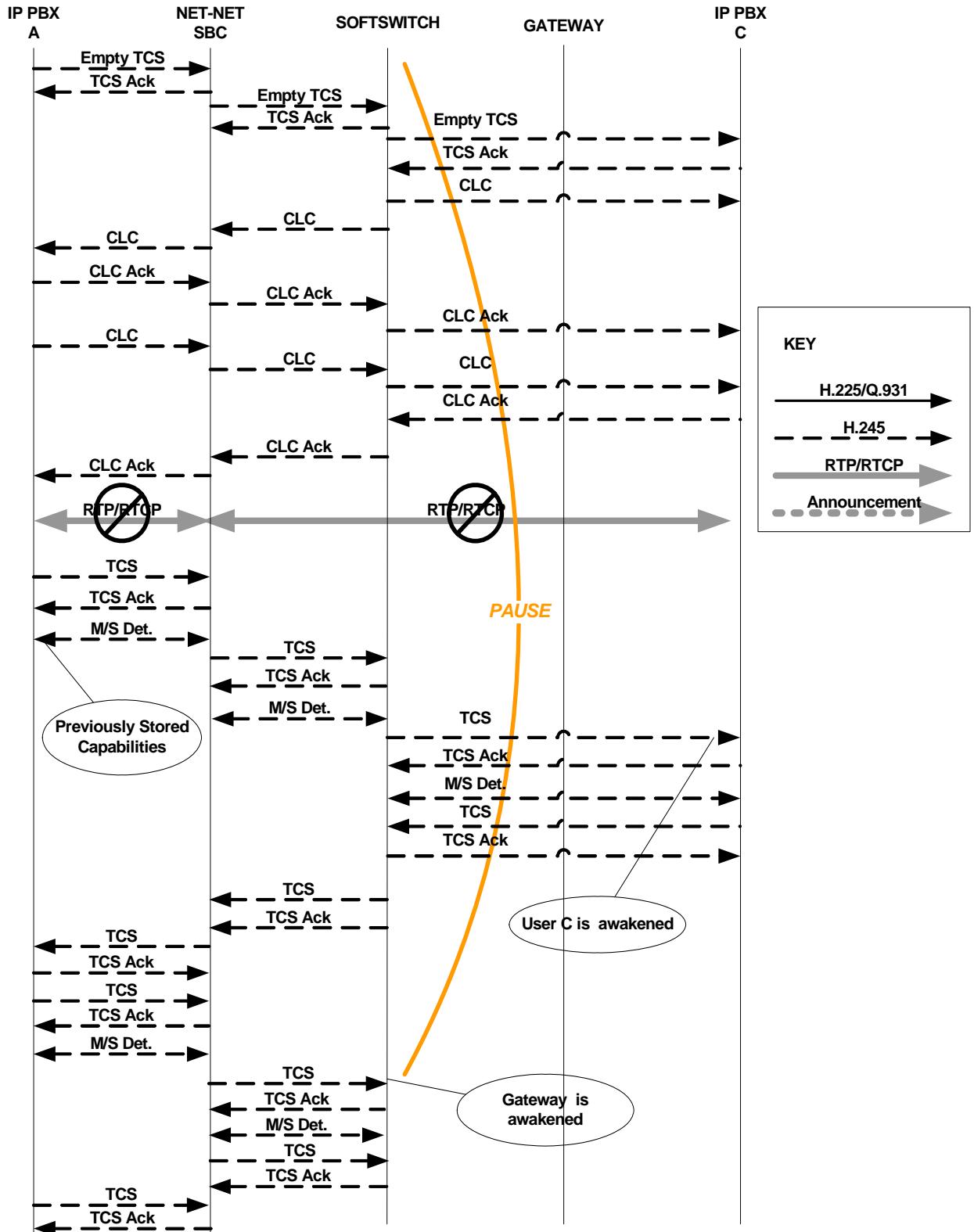


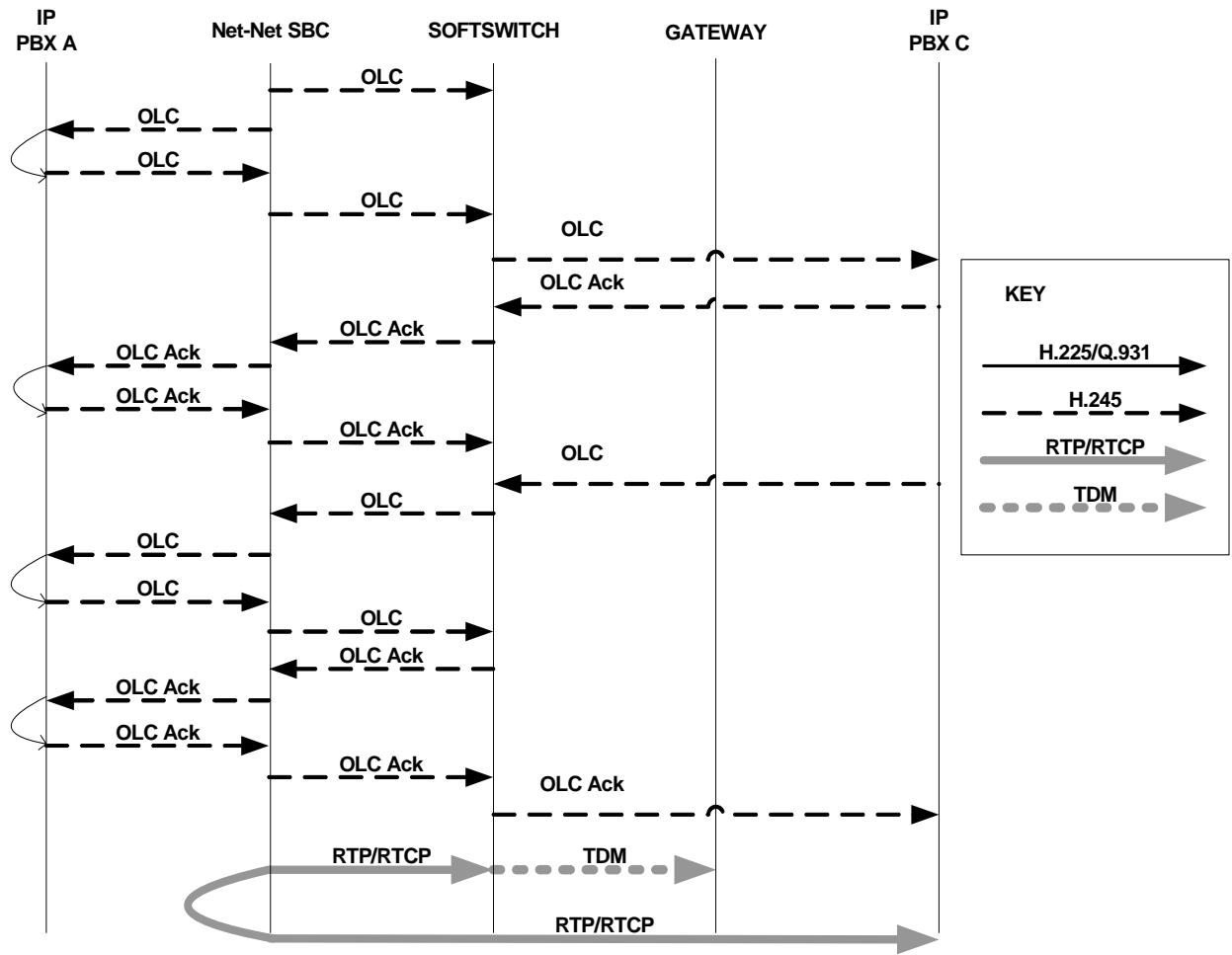
Call Hold and Transfer: Transfer

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 becomes 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

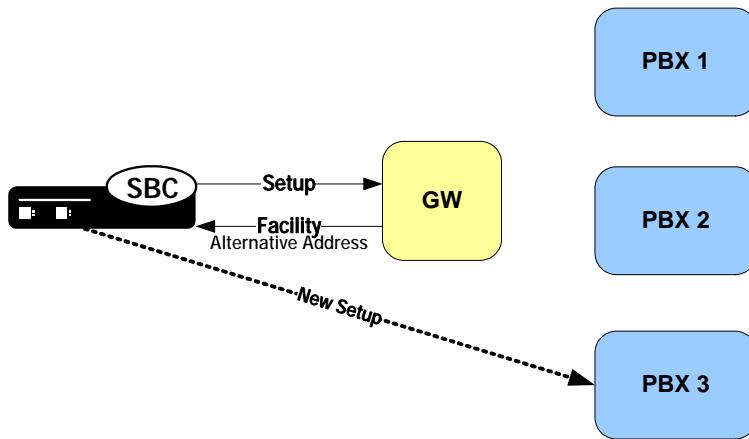
Prior to Release 4.1, the Net-Net SBC did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

- 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.



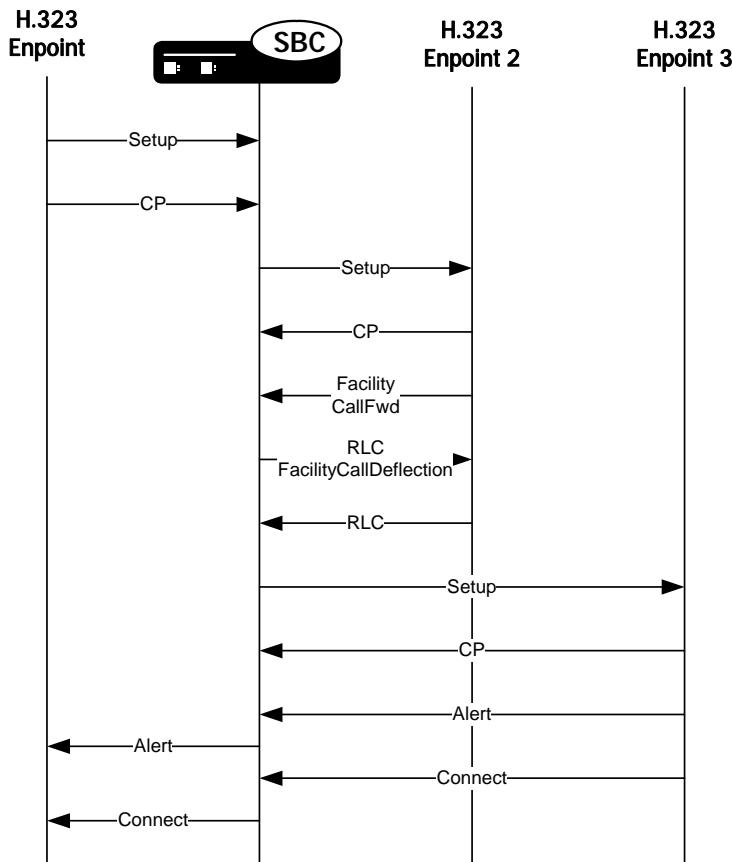
How It Works

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.

H.323: H.239 Support for Video+Content

The Net-Net SBC supports multiple media streams for the same payload, generic capabilities, and H.239 generic messages. As a result, these additions broaden the Net-Net SBC's support for videoconferencing, and free you from having to configure media profiles for H.323 support.

Note: These additions are supported for H.323-H.323 traffic only. These additions do not support SIP-H.323 interworking (IWF), so you still need to configure media profiles for that application.

Multiple Media Streams with the Same Payload

In releases prior to S-C6.2.0, the Net-Net SBC supports multiple audio-video-data streams only if those streams use different payload types. The Net-Net SBC's behavior is extended to provide this support as of Release S-C6.2.0. The Net-Net SBC identifies extendedVideoCapability used to establish an additional channel for H.239-compliant endpoints, an OLC that was formerly not supported.

Support for Generic Capabilities

This feature identifies the OIDs shown in the table below and uses the dynamicPayload type to from the incoming OLC to generate its own OLC. So you no longer need media profiles for: genericAudio, genericVideo, and genericData.

Capability Name	Capability Class	Capability Identifier
H.283	Data protocol	{itu-t (0) recommendation (0) h (8) 283 generic-capabilities (1) 0}
G.722.1	Audio protocol	{itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) 0}
G.722.1 Extension	Audio protocol	{itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) extension (1) 0}
H.324	Data protocol	{itu-t (0) recommendation (0) h (8) 324 generic-capabilities (1) http (0)}
H.263	Video protocol	{itu-t (0) recommendation (0) h (8) 263 generic-capabilities (1) 0}
		Note: Use of this capability to signal H.263 "Profiles and Levels" per Annex X/H.263 should always be accompanied in parallel by the signalling of the same modes in H263VideoCapability. This is necessary to ensure that systems which do not recognize the H.263 generic capabilities continue to interwork with newer systems.
H.224	Data protocol	{itu-t (0) recommendation (0) h (8) 224 generic-capabilities (1) 0}
G.722.2	Audio protocol	{itu-t (0) recommendation (0) g (7) 7222 generic-capabilities (1) 0}
G.726	Audio protocol	{itu-t (0) recommendation (0) g (7) 726 generic-capabilities (1) version2003 (0)}
H.241/H.264	Video protocol	{itu-t (0) recommendation (0) h (8) 241 specificVideoCodecCapabilities (0) h264 (0) generic-capabilities (1)}

Capability Name	Capability Class	Capability Identifier
H.241/H.264	Video protocol	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPPacketization(0) RFC3984NonInterleaved(1)}
H.241/H.264	Video protocol	{itu-t(0) recommendation(0) h(8) 241 specificVideoCodecCapabilities(0) h264(0) iPPacketization(0) RFC3984Interleaved(2)}

Support for H.239 Generic Messages

This section describes the Net-Net SBC's support for H.239 Generic Messages.

Generic Message	Description
Generic Request	<ul style="list-style-type: none"> flowControlReleaseRequest—Used when a device wants to add a channel toward an MCU that has sent multipointConference, or if the device wants to increase a channel bit rate when the channel is flow-controlled. The message has the channelId, which is the logicalChannelNumber of the channel. The Net-Net SBC proxies this message, replacing the channelId with the logicalChannelNumber of its channel. presentationTokenRequest—Request by the sender to acquire the indicated token. The message has the channelId, which is the logicalChannelNumber of the channel. The Net-Net SBC proxies this message, replacing the channelId with the logicalChannelNumber of its channel.
Generic Response	<ul style="list-style-type: none"> flowControlReleaseResponse—Sent in response to the flowControlReleaseRequest, either acknowledging or rejecting the request. The “acknowledge” response indicates the far-end device intends to make a best-effort attempt to comply with the request. The exact bit rate requested may not be allocated. The “reject” response indicates that the far-end device does not intend to comply with the request. The response contains the channelId that was sent in the request. While proxying the response, the Net-Net SBC will replace the channelId with the channelId it received in the request. presentationTokenResponse—Sent in response to the presentationTokenRequest. The response will either confirm or reject the assignment of the indicated token to the sender of the presentationTokenRequest. The response contains the channelId that was received in the request. While proxying the response, the Net-Net SBC will replace the channelId with the channelId it received in the request.
Generic Command	<ul style="list-style-type: none"> presentationTokenRelease—Sent by the device holding the token in order to relinquish the token. The message has the channelId, which is the logicalChannelNumber of the channel. The Net-Net SBC proxies this message, replacing the channelId with the logicalChannelNumber of its channel.
Generic Indication	<ul style="list-style-type: none"> presentationTokenIndicateOwner—Indicates who owns the token. The message has the channelId, which is the logicalChannelNumber of the channel. The Net-Net SBC proxies this message, replacing the channelId with the logicalChannelNumber of its channel.

**Support for
Miscellaneous
Indication**

An endpoint sends a miscellaneous indication to send (logicalChannelActive) or stop (logicalChannelInactive) live video streams. The message has a channelId, which is the channel's logicalChannelNumber. The Net-Net SBC proxies this message, replacing the channelId with the logicalChannelNumber of its own channel.

ACLI Signaling Mode Configuration Examples

The following ACLI displays provide examples of the [Signaling Modes of Operation \(511\)](#) 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 a sample for the [Back-to-Back Gateway Signaling \(511\)](#) 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-aliases

                                h323-ID=private
    ras-port        1719
    auto-gk-discovery
    multicast       224.0.1.41:1718
    gatekeeper      x.x.x.x (IP address of GkZone1)
    gk-identifier   gk-zone1.acme.com
    q931-port      1720
    alternate-transport
    q931-max-calls 200
    h245-tunneling
    fs-in-first-msg
    call-start-fast
    call-start-slow
    media-profiles
    process-registration
    anonymous-connection
    proxy-mode
    filename

h323-stack
    name          zone2
    state         enabled
    isgateway     enabled
    realm-id     Domainrealm
    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-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	
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 GkZone3)
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	Domainrealm
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	disabled
process-registration	disabled
anonymous-connection	disabled
proxy-mode	
filename	

Back-to-Back Gatekeeper Proxy and Gateway

This example provides a sample for the [Back-to-Back Gatekeeper Proxy and Gateway \(512\)](#) 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	disabled
filename	
h323-stack	
name	zone2
state	enabled
isgateway	disabled
realm-id	DomainCreate
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-aliases	
ras-port	h323-ID=acme01
auto-gk-discovery	1719
multicast	disabled
gatekeeper	0.0.0.0:0
gk-identifier	x.x.x.x(IP address of GkZONE2)
q931-port	gk-zone2.acme.com
alternate-transport	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	
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-aliases	
ras-port	h323-ID=private
auto-gk-discovery	1719
multicast	disabled
gatekeeper	0.0.0.0:0
gk-identifier	x.x.x.x(IP address of GkZone3)
q931-port	gk-zone3.acme.com
	1720

```

al ternate-transport
q931-max-call-s          200
h245-tunneling
fs-in-first-msg
call-start-fast
call-start-slow
media-profiles
process-registration
anonymous-connection
proxy-mode
filename

h323-stack
    name           zone4
    state          enabled
    isgateway      disabled
    realm-id      DomainCreate
    assoc-stack
    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
    multicast      0.0.0.0:0
    gatekeeper     x.x.x.x(IP address of GkZone4)
    gk-identifier   gk-zone4.acme.com
    al ternate-transport
    q931-port      1720
    q931-max-call-s 200
    h245-tunneling
    fs-in-first-msg
    call-start-fast
    call-start-slow
    media-profiles
    process-registration
    anonymous-connection
    proxy-mode
    filename

```

Interworking Gatekeeper-Gateway

This example provides a sample for the [Interworking Gatekeeper-Gateway \(513\)](#) 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-aliases	
	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
is-gateway	enabled
realm-id	DomainCreate
assoc-stack	zone1
local-ip	x. x. x. x(IP address of VGW-C)
max-calls	200
max-channels	10
registration-ttl	0
terminal-aliases	
	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
is gateway	disabled
real m-id	zone3real m
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
is gateway	enabled
real m-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 \(521\)](#).

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

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
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
19	reserved	A	
20	unassigned	A	
21	unassigned	A	
22	unassigned	A	
23	unassigned	A	
dyn	GSM-HR	A	8000
dyn	GSM-EFR	A	8000
dyn	L8	A	var.
dyn	RED	A	
dyn	VDVI	A	var.
24	unassigned	V	
25	CelB	V	90000
26	JPEG	V	90000
27	unassigned	V	
28	nv	V	90000
29	unassigned	V	
30	unassigned	V	
31	H261	V	90000
32	MPV	V	90000
33	MP2T	AV	90000
34	H263	V	90000
35-71	unassigned	?	
72-76	reserved for RTCP conflict avoidance	N/A	N/A
77-95	unassigned	?	

Payload Type	Encoding Name	Audio (A)/Visual (V)	Clock Rate (Hz)
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 \(533\)](#), the Net-Net SBC modifies and deletes certain fields as outlined 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

Field Name	Message	Deleted	Modified	Value Used in Modification
endpointAlias.transportID	URQ	X		
endpointAliasPattern.Wwildcard.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		
preGrantedARQ.useSpecifiedTransport	RCF	X		
rasAddress	RRQ		X	Mapped virtual RAS address allocated by the Net-Net SBC for registering endpoint
remoteExtentsionAddress.transportID	ARQ, ACF	X		
srcCallSignalAddress	ARQ	X		
srcInfo.transportID	ARQ	X		
supportedH248Packages	RRQ	X		
supportsAltGK	RRQ	X		
supportedPrefixes.prefix.transportID	RCF, URQ	X		
terminalAlias.transportID	RRQ	X		
terminalAliasPattern.wilcard.transportID	RRQ	X		
willRespondToIIRR	RCF, ACF	X		

Field Name	Message	Deleted	Modified	Value Used in Modification
willSupplyUUIEs	RRQ, ARQ			
uuiiesRequested	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

Introduction

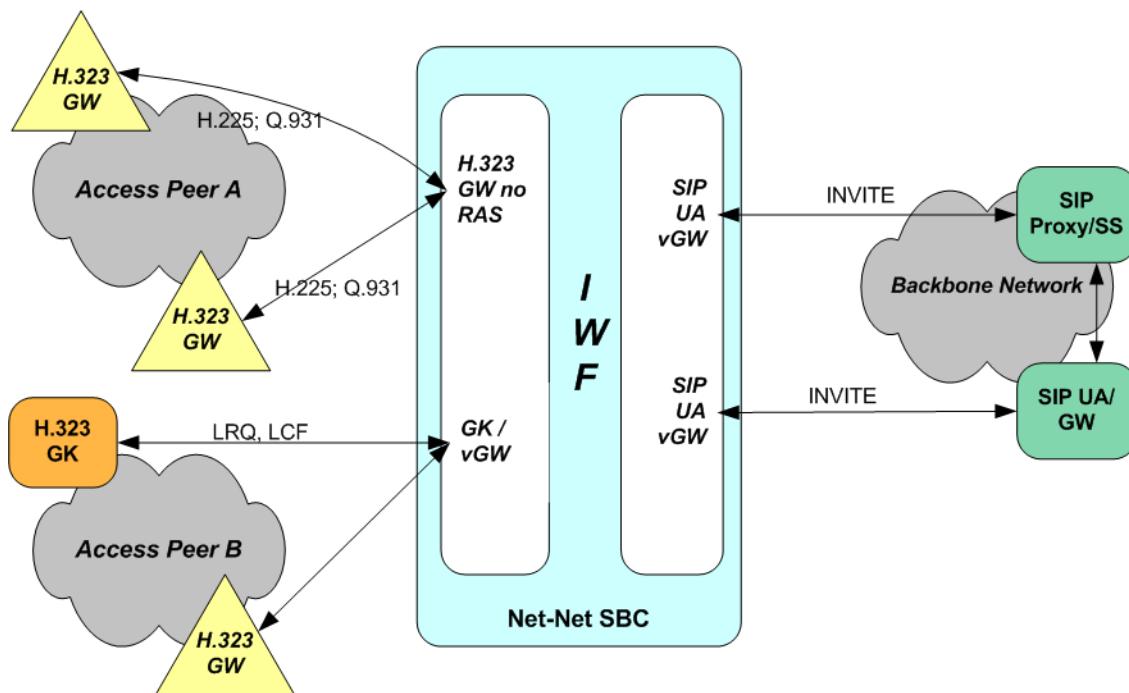
Using the Net-Net SBC's interworking (IWF) function, you can interconnect SIP networks with H.323 networks. Considering the large amount of H.323 deployments already in place and the continuing emergence of SIP in new VoIP deployments, the IWF provides a much-needed solution. SIP providers can maintain a single-protocol backbone while exchanging VoIP sessions with H.323 providers.

The [H.323 Signaling Services \(509\)](#) 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

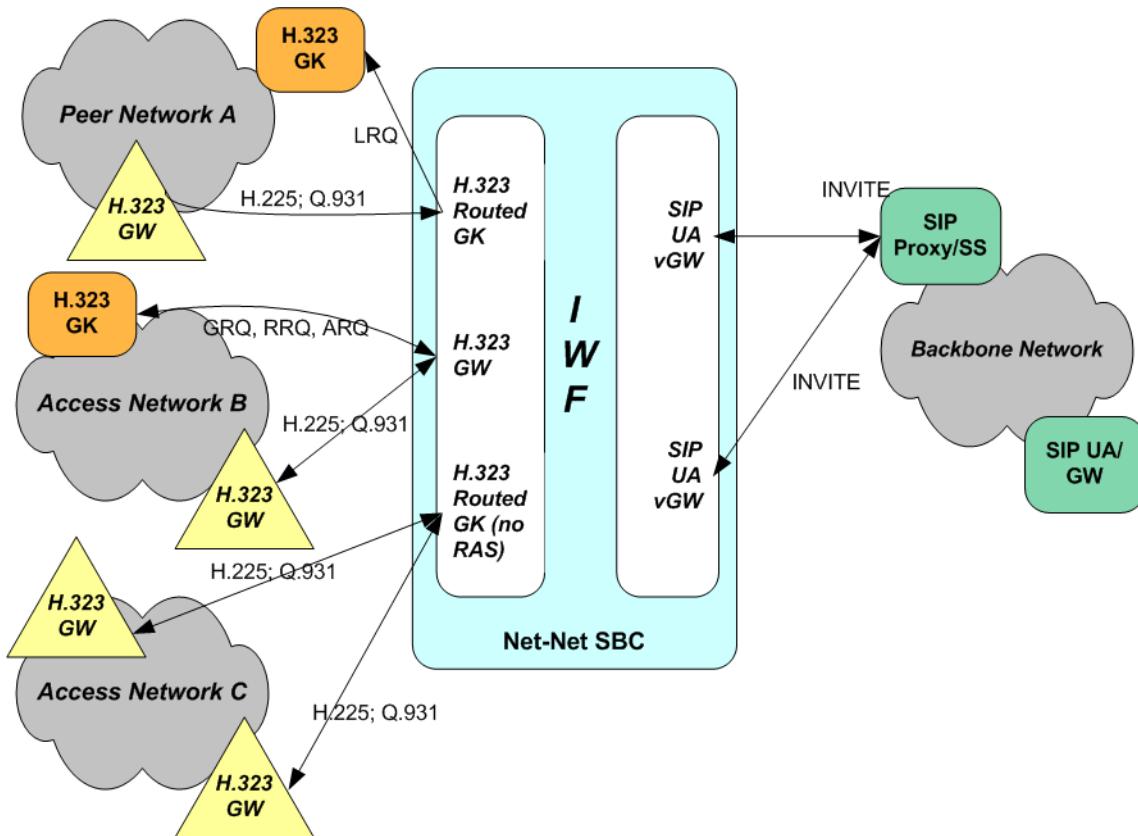
Access Network Application

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.

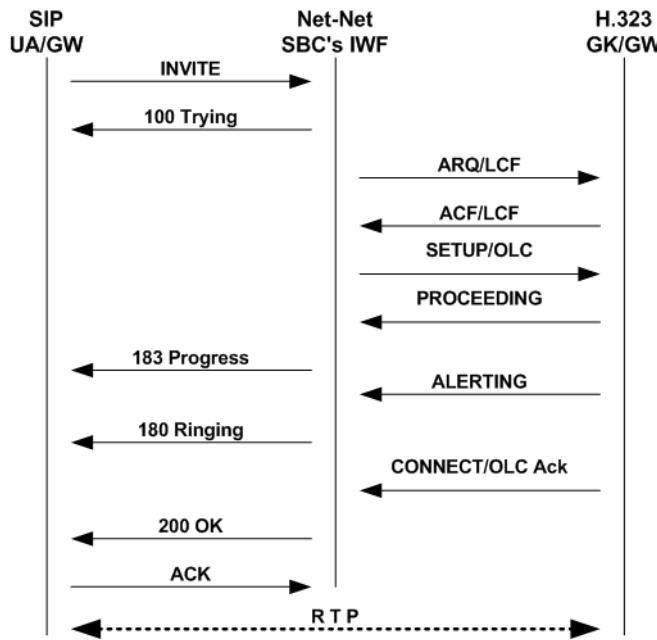
This section's call flow diagrams show how SIP and H.323 messages flow between SIP and H.323 devices, with the Net-Net SBC positioned between the two entities so it can perform translations. The following two sample scenarios with Fast Start appear in the diagrams below, although other scenarios are 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.

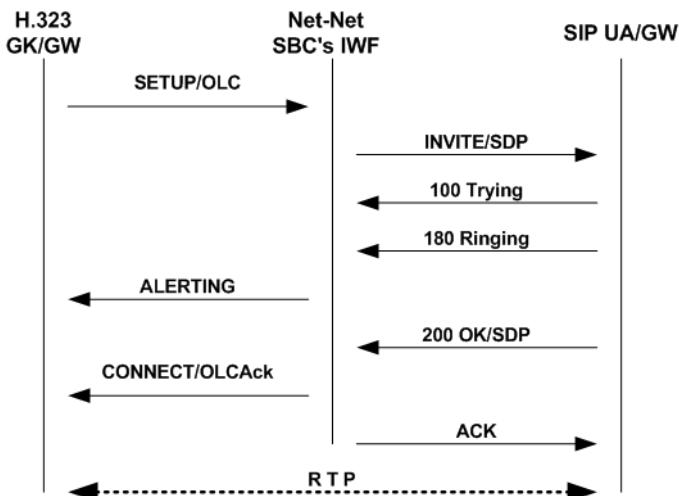
Once the H.323 endpoint responds with an ACF or LCF, the Net-Net SBC reissues the SIP INVITE on the H.323 side as an H.225 Setup, which is sent with the OLC. Then the H.323 endpoint responds with Proceeding and Alerting messages (which correspond respectively to SIP 183 Progress and 180 Ringing messages). 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.

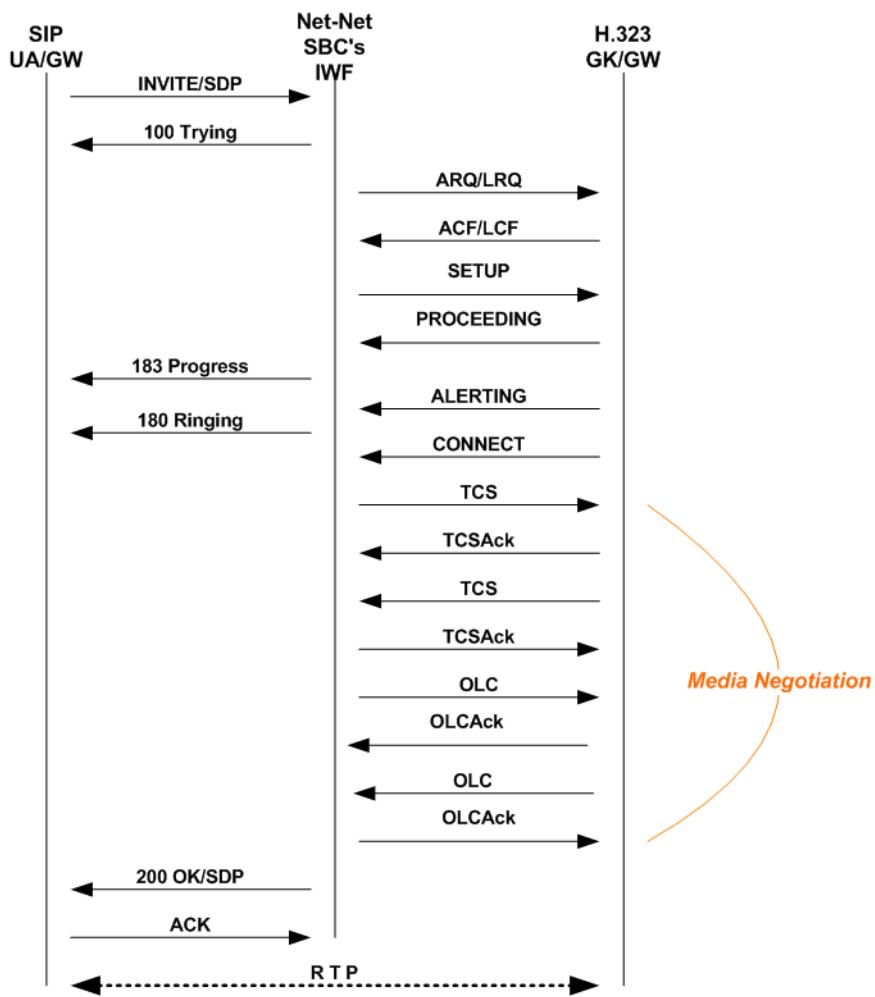
This section’s call flow diagrams show how SIP and H.323 messages flow between SIP UA/GW and an H.323 GK/GW, with the Net-Net SBC positioned between the two entities so it can perform translations. Two sample scenarios with Slow Start appear in the diagrams below:

- SIP being interworked to Slow Start H.323
- Slow Start H.323 being interworked to SIP

H.323 SIP to Slow Start

In the following diagram below, a SIP endpoint (such as a UA or a SIP Gateway) initiates a session by sending an INVITE request destined for an H.323 Slow Start endpoint (a GK or GW). Between these entities, the 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 \(597\)](#), 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.

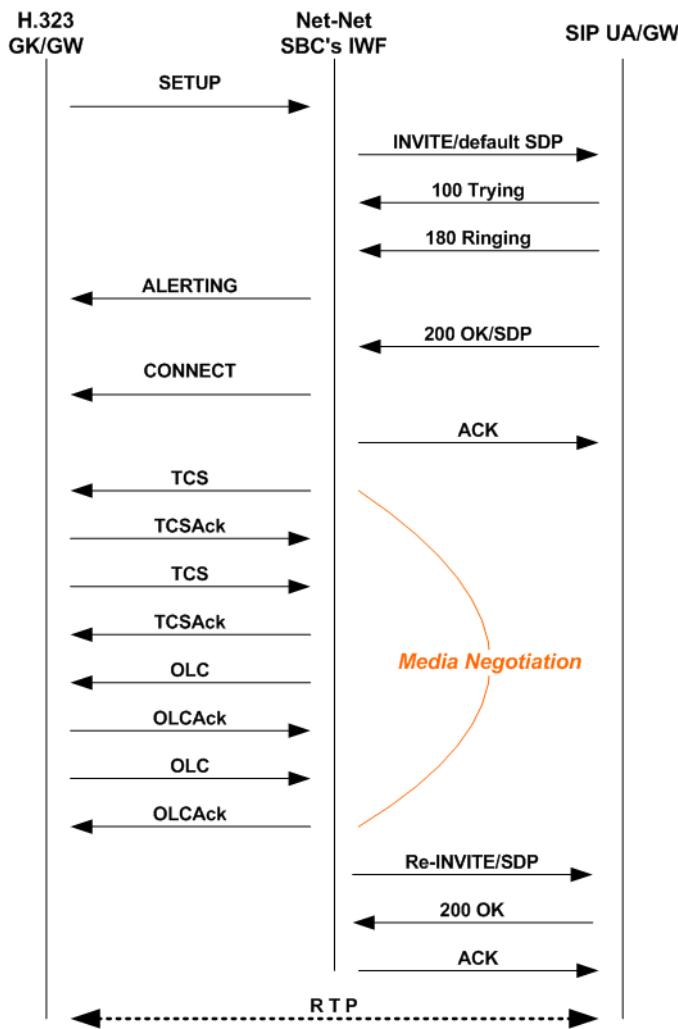


H.323 Slow Start to SIP

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 \(598\)](#), 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
Invalid Revision	505 Version Not Supported
No Permission	401 Unauthorized

H.323 Disconnect Reason	SIP Status
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 a call that 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

SIP Status	H.323 Release Complete Reason
604 Does Not Exist Anywhere	Unreachable Destination
606 Not Acceptable	Undefined Reason

Q.850 Cause to H.323 Release Complete Reason

When a call that requires the IWF terminates from the H.323 side and no H.323 Release Complete Reason is specified, the 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 \(604\)](#) 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

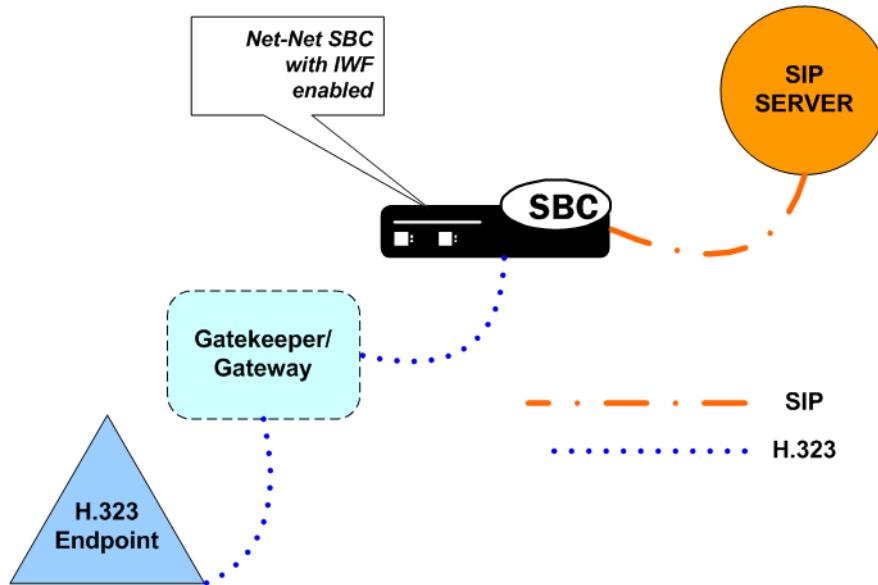
H.245 Type	SDP Media Type
g728	G728
g729wAnnexB	G729
g729	G729 fmtp:18 annexb=no
h261VideoCapability	H261
h263VideoCapability	H263

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

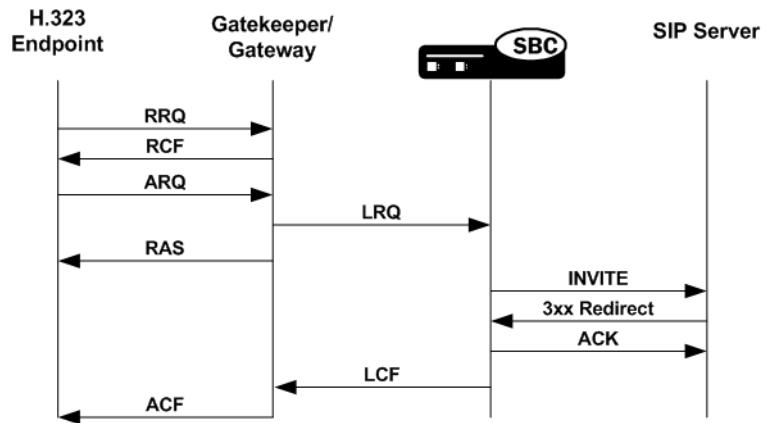
When it needs to interact with a SIP Redirect server, the Net-Net SBC can interpret the SIP messages and manage them on the H.323 side of the session. For IWF sessions, the Net-Net SBC handles SIP Redirect and H.323 LRQ messages.



Redirect—LRQ Management Sample 1

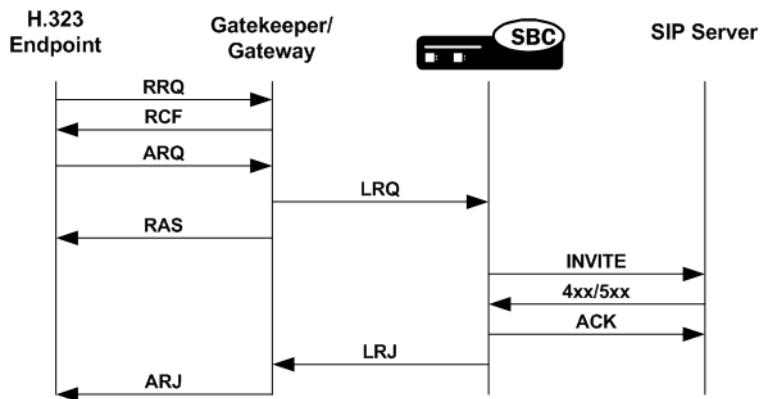
This section presents three possible scenarios for SIP Redirect—H.323 LRQ management.

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.



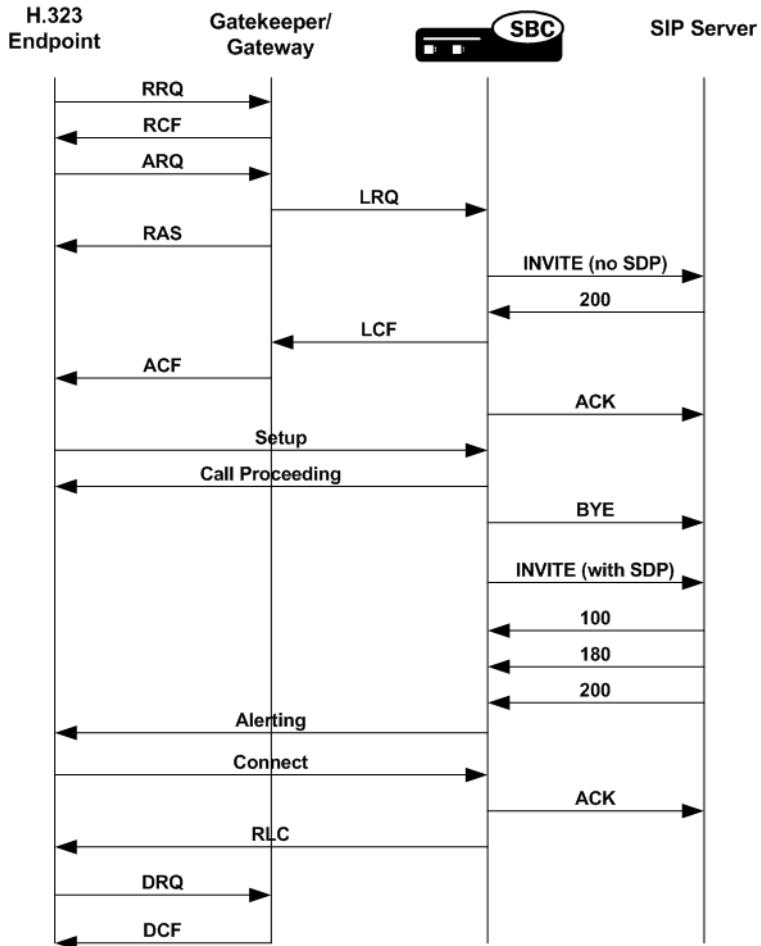
Redirect—LRQ Management Sample 2

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 for 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.

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 that requires the IWF when the type of media being sent while a session is in progress changes. For example, a fax transmission might require mid-session media change; besides fax, other

applications of this feature are possible. To support the transmission of a T.38 fax sent over an IWF session, some media channels must be opened and others closed. In addition, the 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.

Enhanced Support for FAX Calls

The Net-Net SBC now supports T.38 fax calls in networks containing elements that do not comply with the ITU-T H.323 Annex D recommendation for how to replace an existing audio stream with a T.38 fax stream. This support applies to signaling that requires interworking between SIP and H.323.

In the standard call model following the ITU-T recommendation, the endpoint detecting the fax tone sends an H.245 RequestMode message to its peer with a T.38 data mode. The receiving endpoint returns a RequestMode Ack by way of acknowledgement, triggering the sending endpoint to close its audio channel and open a T.38 fax channel. The receiving endpoint closes and opens the same channels on its end. T.38 fax streams flow upon the acknowledgement of all relevant channels.

However, certain endpoints close their logical channel before sending the H.245 RequestMode message for T.38, leaving the Net-Net SBC with its audio channel still open and without having attempted to open a T.38 fax channel. To overcome this issue, the Net-Net SBC now checks whether or not audio channels have been closed whenever it receives an H.245 RequestMode message for T.38. If it finds a closed audio channel, the Net-Net SBC checks for the presence of a matching outgoing audio channel. A match causes the Net-Net SBC to close the audio channel and continue with the procedure for converting to T.38 fax.

Early Media

For calls 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 that 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.

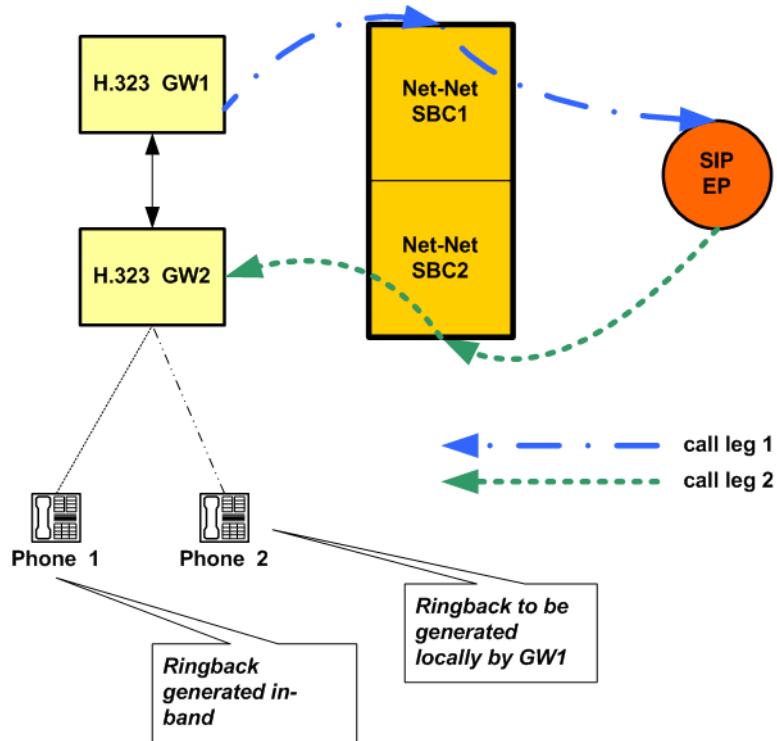
IWF Ringback Support

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
183 with SDP	Progress with PI
180 w/o SDP	Alert without PI
180 with SDP	Alert with PI

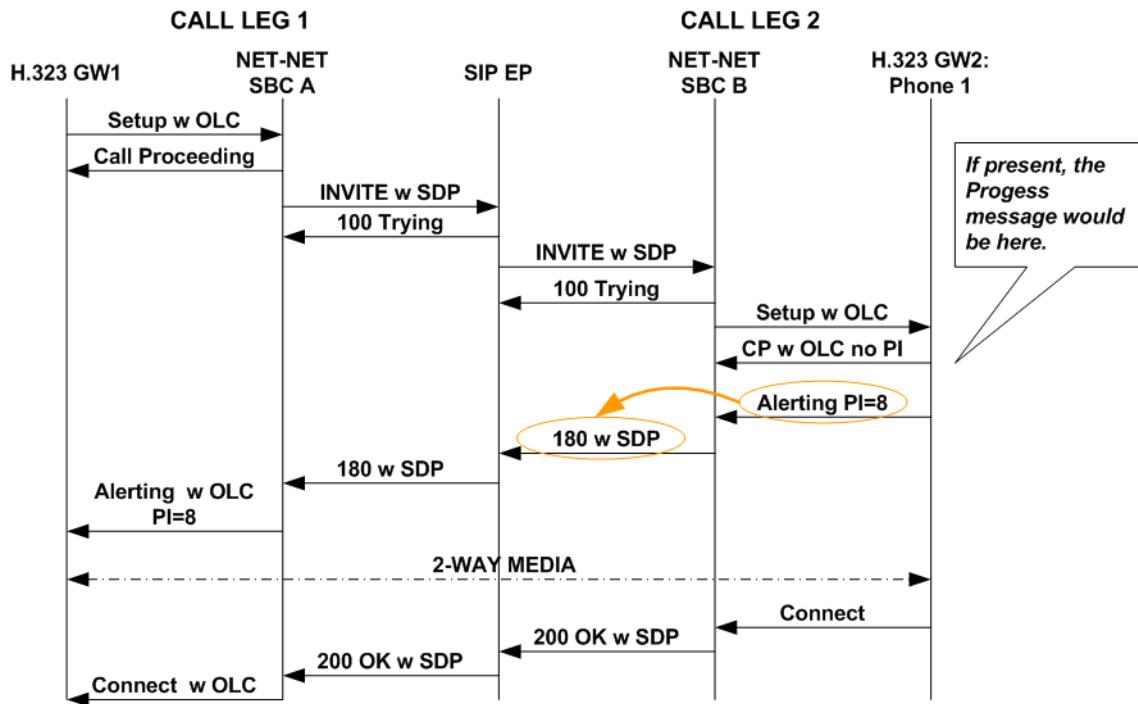
In the following diagram, a call that requires the IWF passes through the Net-Net SBC twice, creating two call legs. The call originates from H.323 GW1 and terminates in Phone 1 or Phone 2.



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.

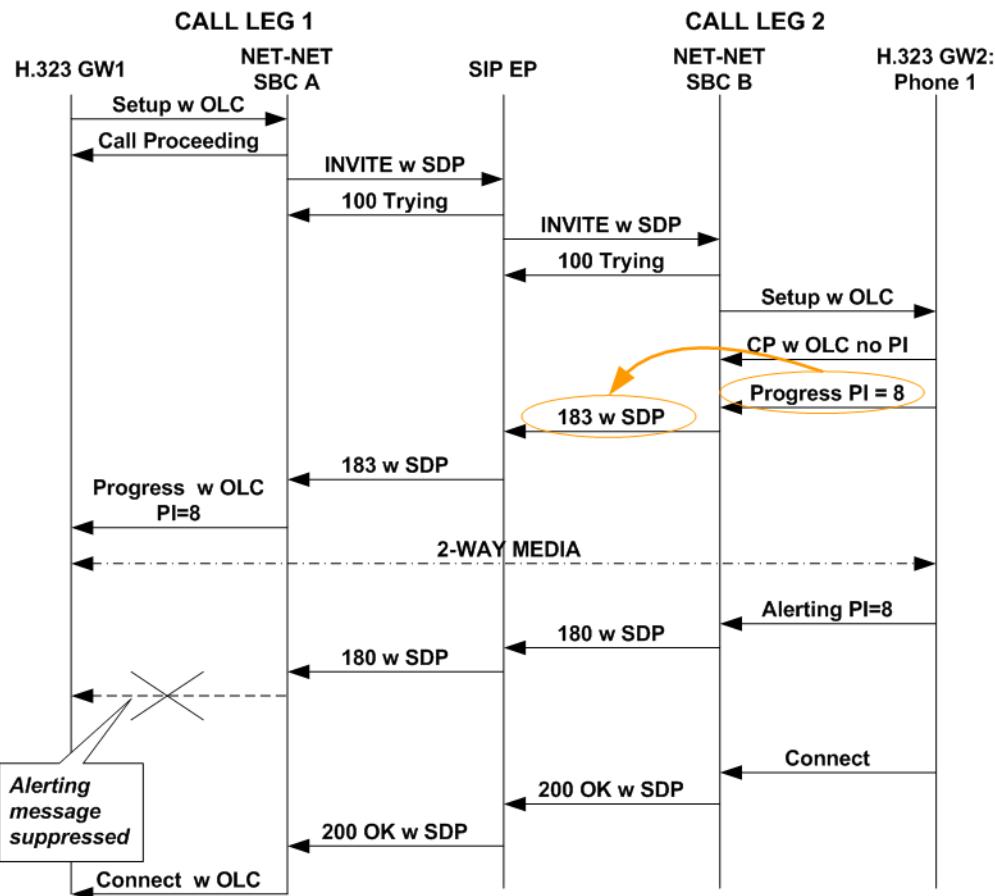
In this diagram, you can see that the Net-Net SBC maps the progress indicator included in the Alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. When the Progress message appears, it contains the progress indicator rather than the Alerting message containing it.



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.

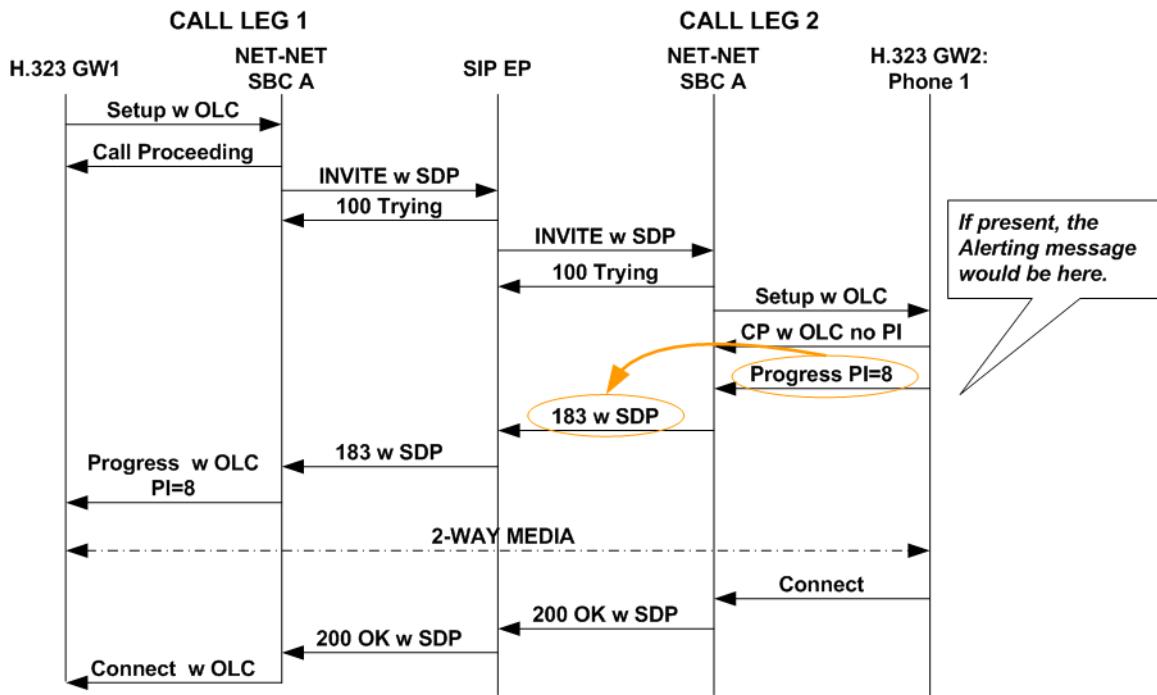
For this call flow, you can see again that the Net-Net SBC maps the progress indicator included in the alerting message sent from Phone 1 through H.323 GW2 to a SIP 180 message with SDP. Note that now the Progress message contains the progress indicator.



Sample 3: In-band Ringback without Alerting Message

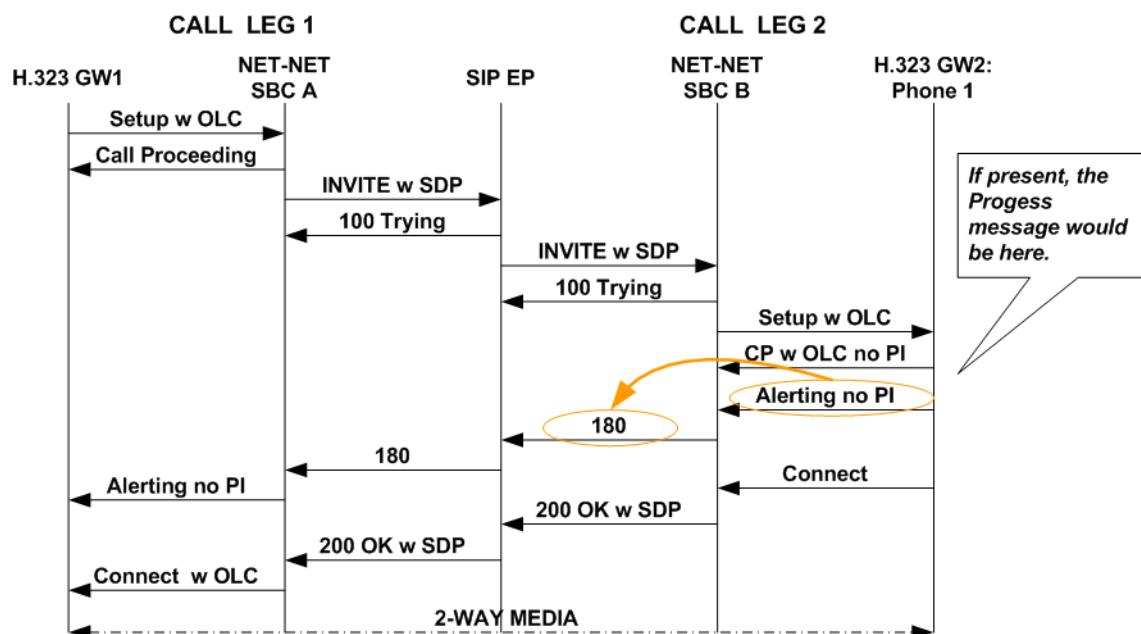
This sample flow shows how the Net-Net SBC handles a call that requires the IWF where there is no progress message. In this call flow, there is a progress indicator of eight (8), meaning that ringback is in-band.

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.



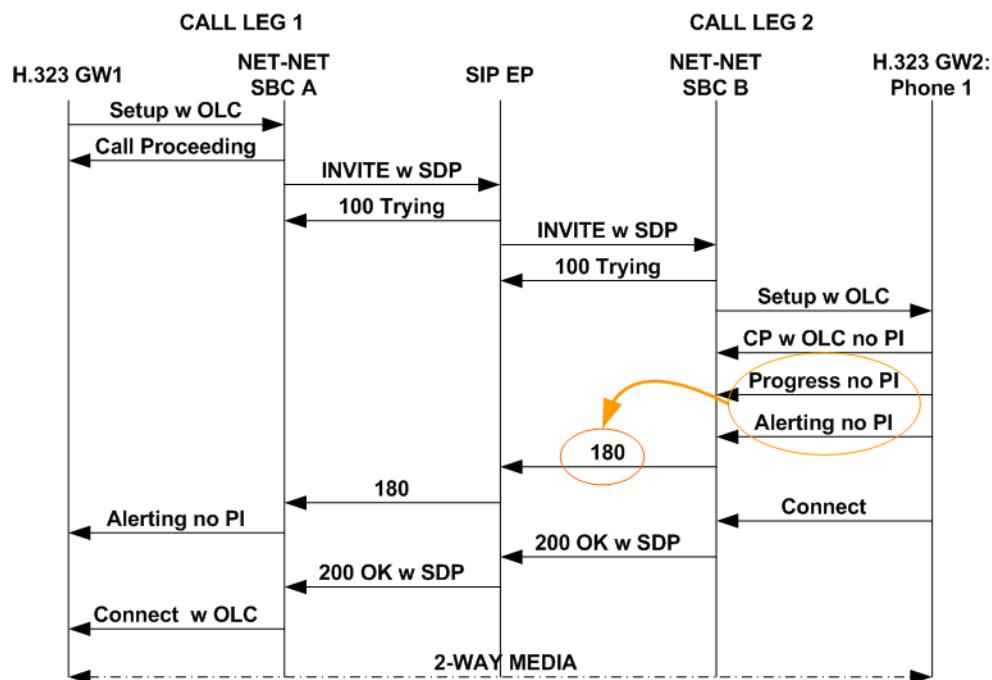
Sample 4: Out-of-band Ringback without Progress Message

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.

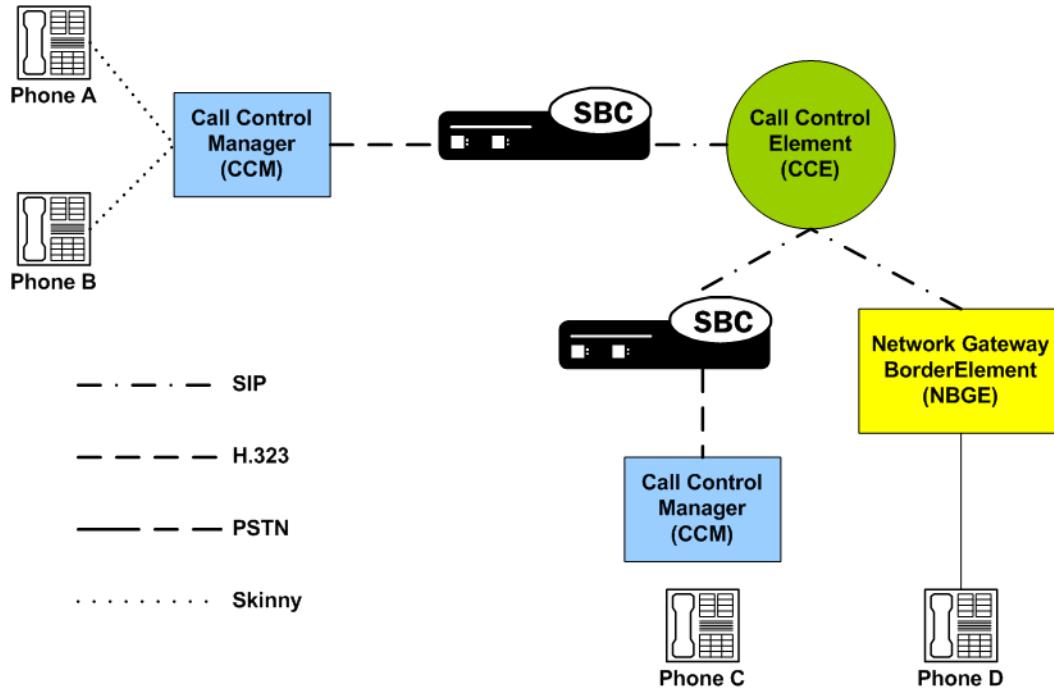


H.323 Endpoint-Originated Call Hold and Transfer

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.

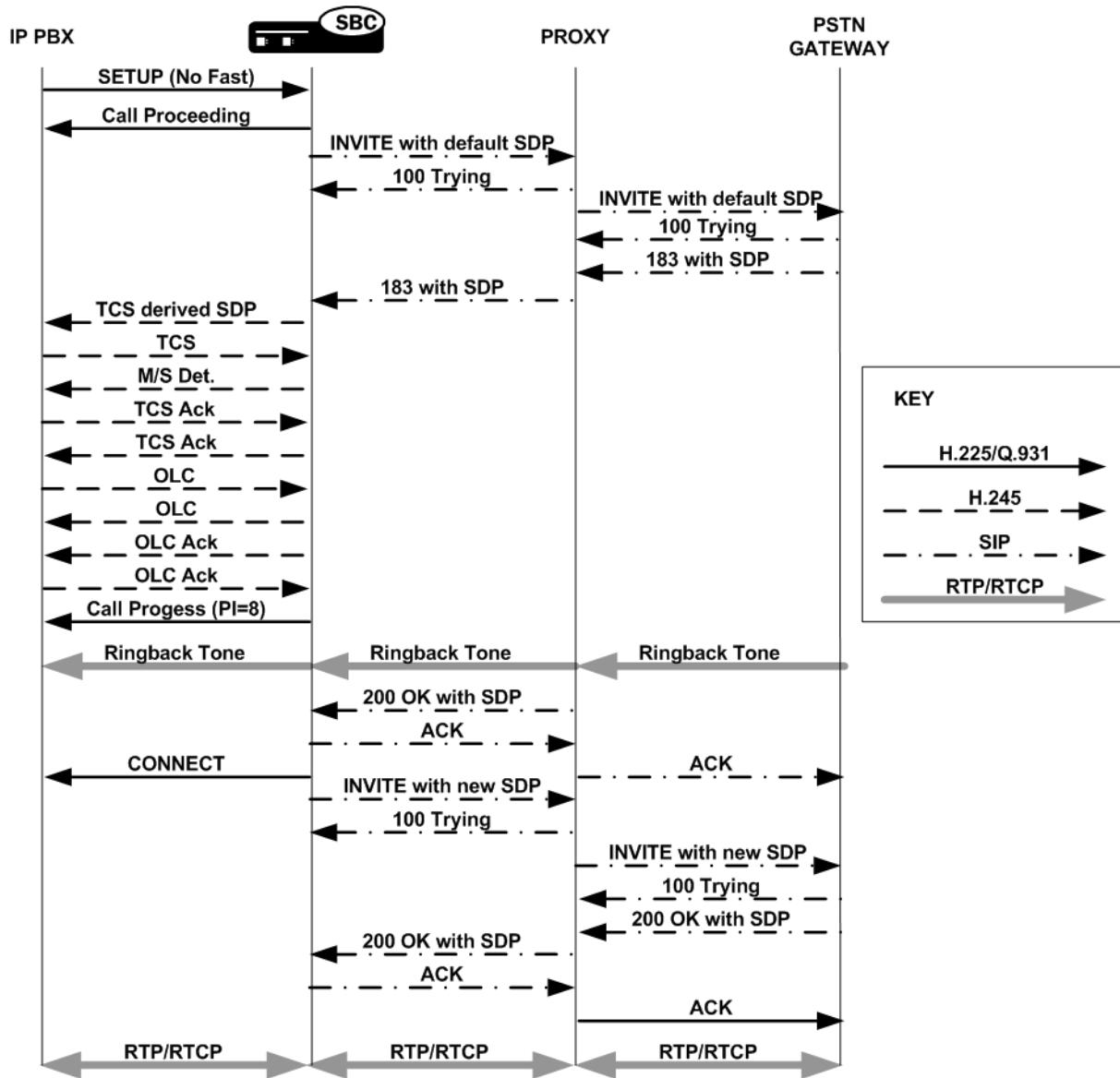
The following diagram shows how the Net-Net SBC provides call hold and transfer support for IWF sessions that originate in H.323. As you review this section’s call flow diagrams, you might want to refer back to the following logical diagram directly below to review the network elements involved, and what protocols they use.



Basic Call

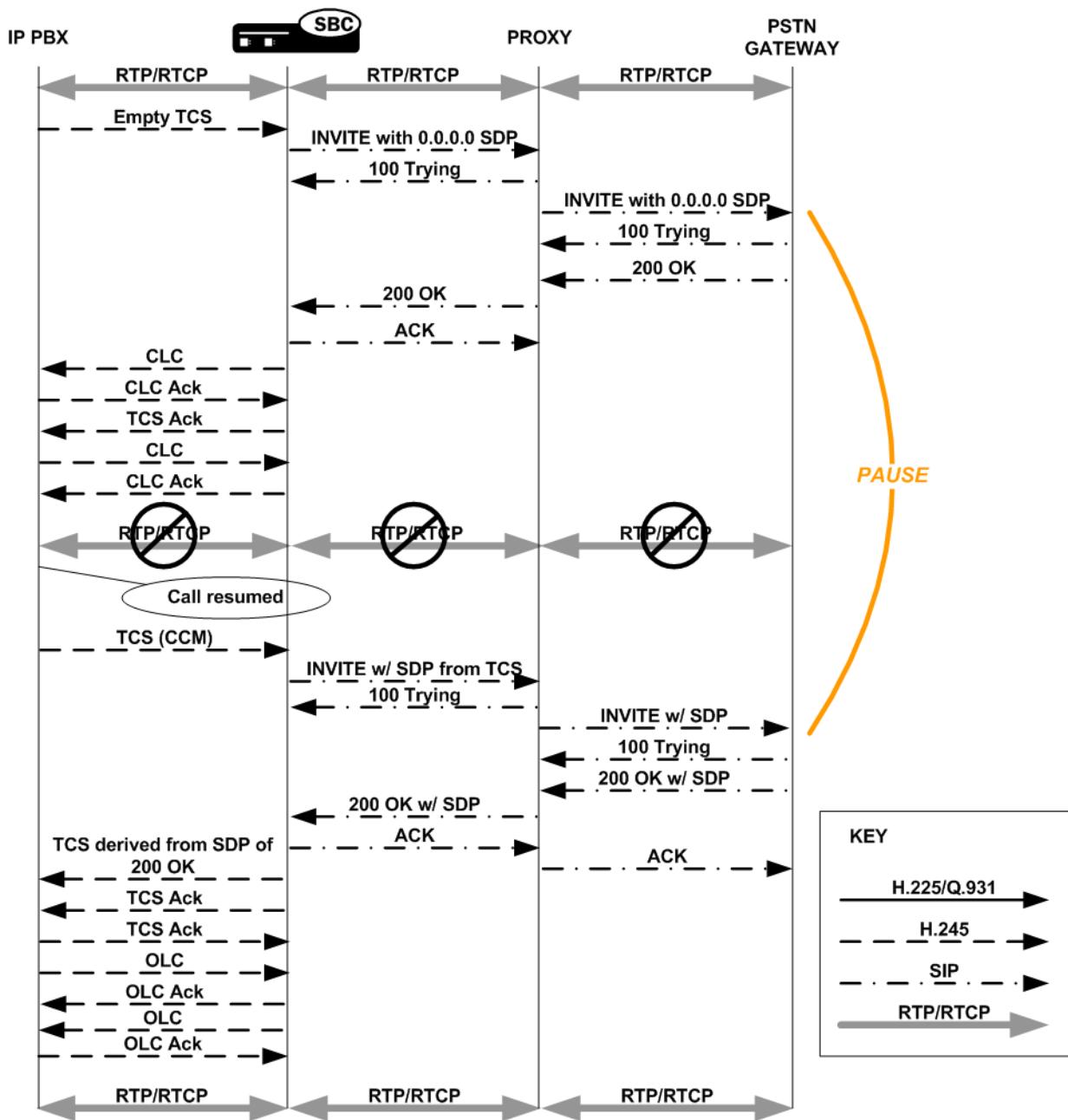
In the following sample basic call, IP PBX A sends an H.323 Slow Starts message ultimately destined for the PSTN through the Net-Net SBC. The Net-Net SBC performs translation to SIP and inserts default information about media. Once the PSTN gateway responds with a 183 containing SDP, the Net-Net SBC 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**

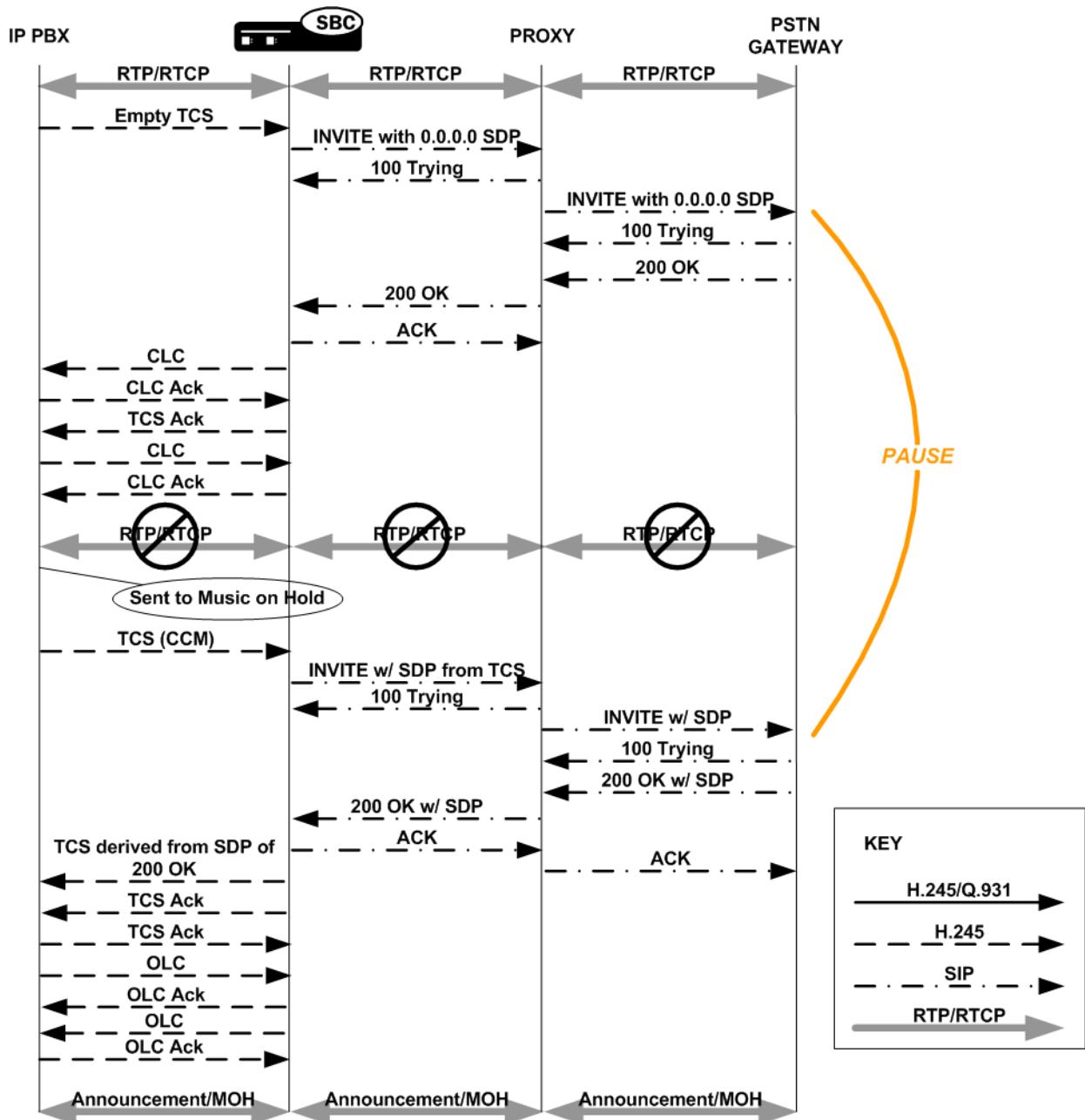
This sample call flow assumes that the IWF call is established and that the RTP/RTCP flow is already in progress. The hold button is pushed, and IP PBX A sends an empty TCS to the Net-Net SBC. The Net-Net SBC puts the called party on hold by sending an INVITE message with 0.0.0.0 SDP to the SIP side 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.



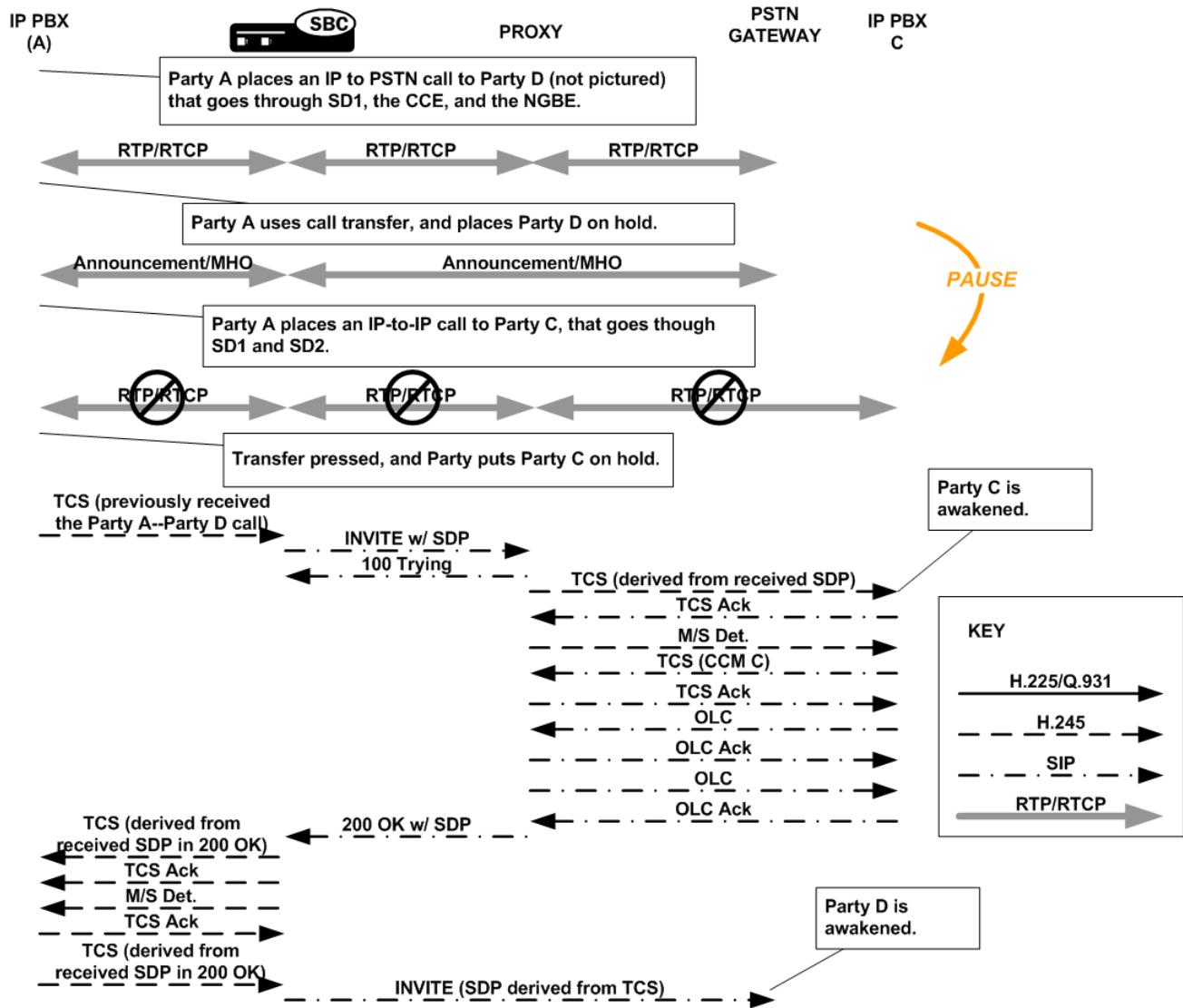
Transfer

The call flow described in this section recalls the diagram at the top of the [H.323 Endpoint-Originated Call Hold and Transfer \(616\)](#) 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.

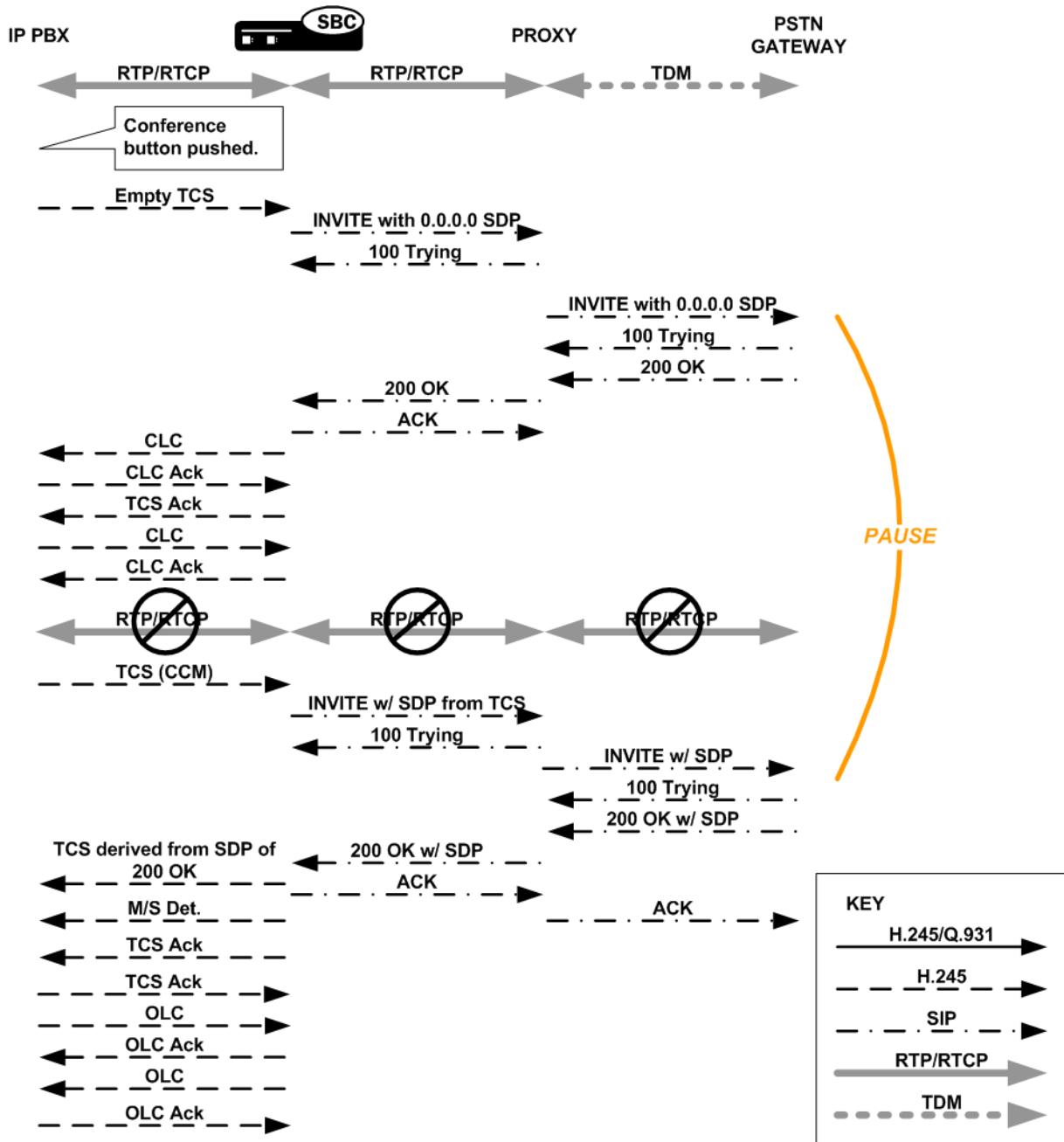
In order to redirect the media so that it flows between Party C and Party D, the Net-Net SBC1 and IP PBX C exchange OLC and OLC Ack messages that contain address information for Party C and for Party D. Address information for both parties is contained in the OLC Ack messages that the Net-Net SBC exchanges 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 that requires the IWF that starts in H.323, the Net-Net SBC uses a scenario much like the one used for holding a call that requires the IWF. Here again, the INVITE with 0.0.0.0 as the media address and the closing of logical channels stops the flow of RTP/RTCP. After signaling and SDP/media information are re-established, RTP/RTCP for the conference flows.



IWF Call Forwarding

This section describes the Net-Net SBC's IWF Call Forwarding feature, which is supported for calls initiated in SIP that require interworking to H.323.

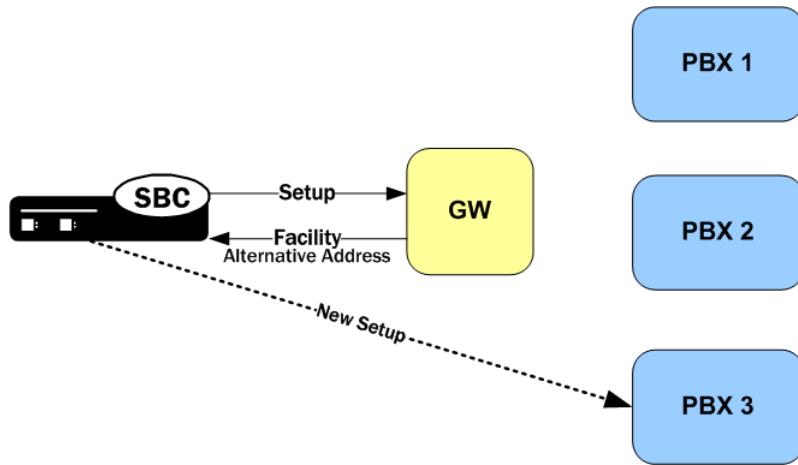
Prior to the implementation of this feature, the Net-Net SBC did not forward calls when the remote H.323 endpoint sent a Facility message with Call deflection as the reason and an alternate address for forwarding. Instead, it would either:

- 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

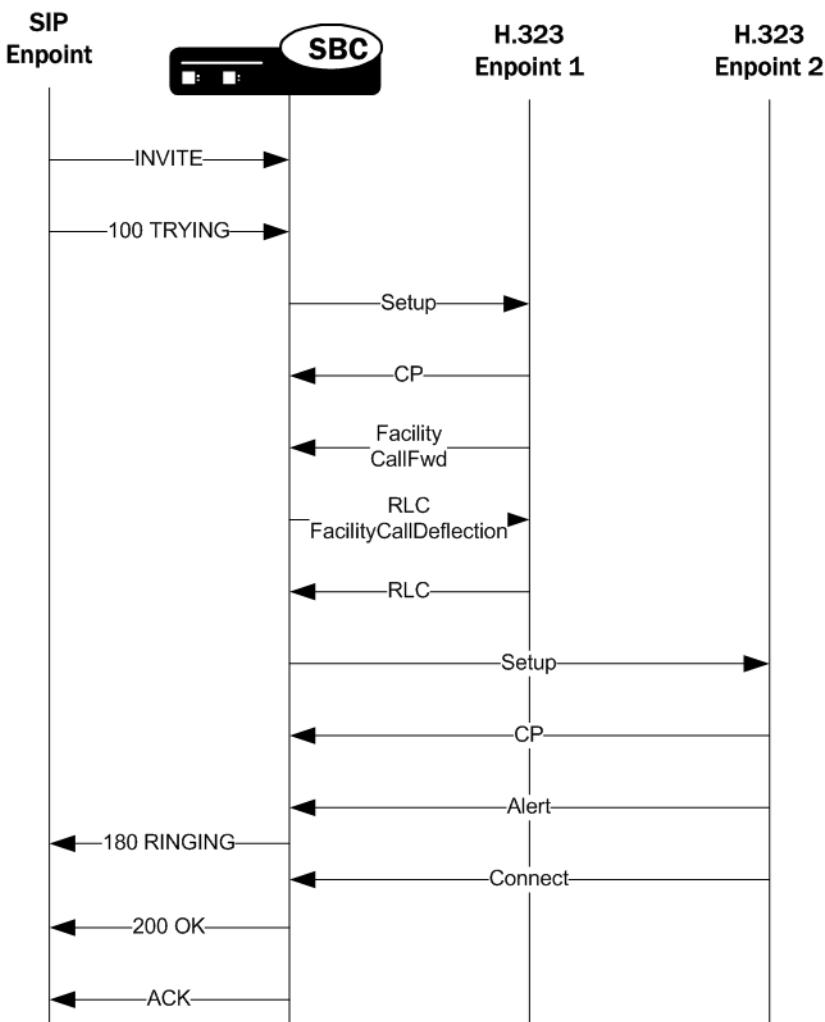
When it receives a Facility message with the reason CallForwarded, the Net-Net SBC looks for an alternate transport address in the Facility's alternativeAddress or alternativeAliasAddress element. The Net-Net SBC releases the egress call with the reason facilityCallDeflection. Then it takes one of two 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 becomes 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 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 \(215\)](#) 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-real *
        state enabled
        last-modified-date 2004-04-20 17:43:13
      policy-attribute
        next-hop sag:sag_internal
        realm internal
        replace-uri-disabled
        carrier
        start-time 0000
        end-time 2400
        days-of-week U-S
        cost 0
        app-protocol SIP
        state enabled
        media-profiles

```

Local Policy in an IWF Session Initiated with H.323

In a session where the 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 \(606\)](#) 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:

<code>iwf-config</code>		
<code>state</code>		<code>enabled</code>
<code>media-profiles</code>		
		<code>PCMU</code>
		<code>telephone-event</code>
<code>logging</code>		<code>disabled</code>

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**—Enable this parameter if you want to translate SIP and H.323 sessions on your Net-Net SBC. The default value is `disabled`. Valid values are:
 - `enabled` | `disabled`
5. **media-profiles**—Enter the name of the media profiles you want to use for IWF translations. This name is either the name of an SDP codec (such as `PCMU`), or it can be `telephone-event` if you are configuring your system for DTMF support.
 If you want to use more than one media profile for SIP/H.323 translations, enter the names in quotation marks with a space between each one.
`ACMEPACKET(iwf-config)# media-profiles "PCMU telephone-event"`

6. **logging**—Enable this parameter if you want the Net-Net SBC to log SIP messages that are related to the IWF. The default value is **disabled**. Valid values are:
 - enabled | disabled

DTMF Support

For calls that require the IWF, you can enable support for the relay of RFC 2833 DTMF digits. The availability of this feature means that the Net-Net SBC is compliant with RFC 2833, which defines two payload formats for DTMF digits. To learn more about this RFC, refer to <http://www.ietf.org/rfc/rfc2833.txt>.

Until the exchange of TCS messages with the H.323 endpoint, the Net-Net SBC has no information about the endpoint's RFC 2833 capabilities. The Net-Net SBC adds *telephone-event* to the SDP on the SIP side of the call.

For calls that require SIP/H.323 translation, you can enable support for the relay of RFC 2833 DTMF digits.

To use this feature, you need to configure a media profile called *telephone-event* and set relevant parameters for it. Application of the media profile can happen either in a session agent configuration or in the IWF configuration.

- The **name** parameter in the media profiles configuration
- The **media-profiles** list in the IWF configuration
- The **media-profiles** list in the session agent configuration

All of the scenarios outlined here assume that you have established a *telephone-event* media profile configuration.

You can configure DTMF support using the following parameters. The way that the Net-Net SBC uses these values is described below. The payload type, part of the media profiles configuration, is dynamic and varies with different endpoints, so there is no default configuration for the *telephone-event* media profile.

The *telephone-event* media profile is used as follows in these types of IWF sessions:

- **Calls that require the IWF originating in H.323 Fast Start**—The Net-Net SBC uses the channels defined in the Fast Start messages to generate SDP on the SIP side of the session.
 - If the incoming H.323 endpoint is an H.323 session agent and the media profiles parameter for the session agent is set to *telephone-event*, the Net-Net SBC will use the information in that media profile to add the *telephone-event* in the SDP.
 - If the incoming H.323 endpoint is not a session agent, the media profile set in the IWF configuration will be used.
- **Calls that require the IWF originating in H.323 Slow Start**—There is no channel (media) information available on the H.323 side.
 - If the incoming H.323 endpoint is configured as a session agent on the Net-Net SBC, then the *telephone-event* parameter in the media profiles set for that session agent configuration will be used in the SDP on the SIP side of the session.
 - If the H.323 endpoint is not a session agent or the *telephone-event* media profile is not configured in the session agent configuration corresponding to the endpoint, then the Net-Net SBC refers to the media profile information configured for the IWF configuration.

- **Calls that require the IWF originating in SIP**—If the TCS was not exchanged before a 200 OK was sent on the SIP side, the Net-Net SBC will behave in one of these two ways.
 - If the outbound H.323 endpoint is configured as a session agent, then the media profiles from that session agent configuration will be used.
 - If the outbound H.323 endpoint is not configured as a session agent, the media profile configured within the IWF configuration with the *telephone-event* value will be used.

As mentioned above, DTMF support is configured by using a combination of the *telephone-event* media profile and either the session agent or IWF configuration. First you set up the media profile, then you apply it to a session agent or to the IWF configuration.

ACLI Instructions and Examples

DTMF support requires you to configure a media profile named *telephone-event*. This section shows you how to set it up.

To configure a *telephone-event* media profile:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure terminal
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# session-router
3. Type **media-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# media-profile
 From this point, you can configure parameters for media profiles. To view see all media profile configuration parameters, enter a ? at the system prompt.
4. **name**—Enter the name **telephone-event** and press <Enter>.
5. **parameters**—Enter the parameters to be applied for the codec; these are the digits that endpoints can support.
6. **media-type**—Leave the default media type set to **audio**.
7. **payload-type**—Set the payload type to **101**, which is the dynamic payload type needed to support this feature.
8. **transport**—Leave the default transport protocol set to **RTP/AVP**.
9. **frames-per-packet**—You can leave this parameter set to **0** (default).
10. **req-bandwidth**—You can leave this parameter set to **0** (default).

Applying the Media Profile

After you have configured the *telephone-event* media profile, you need to apply it either to a H.323 session agent or the global IWF configuration.

To use DTMF support on a per-session-agent basis:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **session-agent**
From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a ? at the system prompt.
4. When you configure a new H.323 session agent, you can configure DTMF support by simply adding the *telephone-event* media profile to the list of media profiles. You can add it along with the other media profiles you might want to use for that session agent.
ACMEPACKET(session-agent)# **media-profiles "telephone-event g711ui aw64k"**

5. When you want to add DTMF support to an H.323 session agent that you have already configured, you need to select that session agent, add the media profile, and save your work.

```
ACMEPACKET(session-agent)# select
<hostname>:
1: 192. 168. 1. 48 real m=          ip=
2: 192. 168. 1. 49 real m=          ip=
3: 192. 168. 1. 50 real m=external ip=
```

selection: 3

```
ACMEPACKET(session-agent)# media-profiles "telephone-event g711ui aw64k"
```

```
ACMEPACKET(session-agent)# done
```

To use DTMF for all IWF translations:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **iwf-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **iwf-config**
From this point, you can configure IWF parameters. To view see all IWF configuration parameters, enter a ? at the system prompt.

4. Add the *telephone-event* media profile to the media profiles list and save your work. If you already have a media profiles for the IWF configuration set up and want to keep them (adding *telephone-event* to the list), then you must type in all of the media profiles that you want to use.

```
ACMEPACKET(i wf-confi g)# medi a-profil es "PCMU telephone-event"
ACMEPACKET(i wf-confi g)# done
```

RFC 2833: DTMF Interworking

This section explains the Net-Net SBC's support of transporting Dual Tone Multi-Frequency (DTMF) in Real-Time Transport Protocol (RTP) packets (as described in RFC 2833) to H.245 User Input Indication (UII) or SIP INFO method interworking.

Multimedia devices and applications must exchange user-input DTMF information end-to-end over IP networks. The Net-Net SBC provides the interworking capabilities required to interconnect networks that use different signaling protocols. Also, the Net-Net SBC provides DTMF translation to communicate DTMF across network boundaries.

The Net-Net SBC supports:

- RFC 2833 to H.245 UII translation for H.323-to-H.323 calls, when one side is a version 4 H.323 device requiring RFC-2833 DTMF event packets, and the other side is a pre-version 4 H.323 device that only uses H.245 UII.
- RFC 2833 to H.245 UII or INFO translation of H.323 to SIP (and SIP to H.323) IWF calls, when one side is a version 4 H.323 device requiring RFC 2833 DTMF event packets and the SIP endpoint only supports INFO messages. Or when one side is a pre-version 4 H.323 device that only uses H.245 UII and the SIP endpoint supports RFC-2833 DTMF event packets.

About RFC 2833

RFC 2833 specifies a way of encoding DTMF signaling in RTP streams. It does not encode the audio of the tone itself, instead a signal indicates the tone is being sent. RFC 2833 defines how to carry DTMF events in RTP packets. It defines a payload format for carrying DTMF digits used when a gateway detects DTMF on the incoming messages and sends the RTP payload instead of regular audio packets.

About H.245 UII

H.245 provides a capability exchange functionality to allow the negotiation of capabilities and to identify a set of features common to both endpoints. The media and data flows are organized in logical channels. H.245 provides logical channel signaling to allow logical channel open/close and parameter exchange operations. The H.245 signaling protocol is reliable, which ensures that the DTMF tones will be delivered.

H.245 User Input Indication (UII) plays a key role in all the services that require user interaction. For video messaging, typical uses of UII include selection of user preferences, message recording and retrieval, and typical mailbox management functions. H.245 UII provides two levels of UII, alphanumeric and signal.

About RFC 2833 to H.245 UII Interworking

The Net-Net SBC provides 2833 to H.245-UII interworking by checking 2833-enabled RTP streams for packets matching the payload type number for 2833. It then sends the captured packet to the host for processing and translation to H.245 UII

messages. A H.245 UII message received by the Net-Net SBC is translated to 2833 packets and inserted into the appropriate RTP stream.

About DTMF Transfer

DTMF transfer is the communication of DTMF across network boundaries. It is widely used in applications such as interactive voice response (IVR) and calling card applications.

The multiple ways to convey DTMF information for packet-based communications include:

- In-band audio: DTMF digit waveforms are encoded the same as voice packets. This method is unreliable for compressed codecs such as G.729 and G.723
- Out-of-band signaling events:
 - H.245 defines out-of-band signaling events (UII) for transmitting DTMF information. The H.245 signal or H.245 alphanumeric methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of through the RTP channel. The tones are transported in H.245 UII messages.

All H.323 version 2 compliant systems are required to support the H.245 alphanumeric method, while support of the H.245 signal method is optional.

- SIP INFO – uses the SIP INFO method to generate DTMF tones on the telephony call leg. The SIP INFO message is sent along the signaling path of the call. Upon receipt of a SIP INFO message with DTMF content, the gateway generates the specified DTMF tone on the telephony end of the call.
- RTP named telephony events (NTE): uses NTE to relay DTMF tones, which provides a standardized means of transporting DTMF tones in RTP packets according to section 3 of RFC 2833.

Of the three RTP payload formats available, the Net-Net SBC supports RTP NTE. NTE is most widely used for SIP devices but is also supported in H.323 version 4 or higher endpoints.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for 2833 telephone event packets using a SDP or H.245 Terminal Capability Set (TCS), which tells the other side what payload type number to use when sending the named event packets to the local device. Most devices use payload type number 101 for 2833 packets, although no default is specified in the standard.

The 2833 packet's RTP header also makes use of the timestamp field. Because events often last longer than the 2833 packets sending interval, the timestamp of the first 2833 packet an event represents the beginning reference time for subsequent 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

See RFC 2833 and [draft-ietf-avt-rfc2833bis-07.txt](#) for more information.

Preferred and Transparent 2833

To support preferred (signaled) 2833 and transparent 2833, the Net-Net SBC provides 2833 detection and generation (if necessary) when the endpoint signals support for 2833.

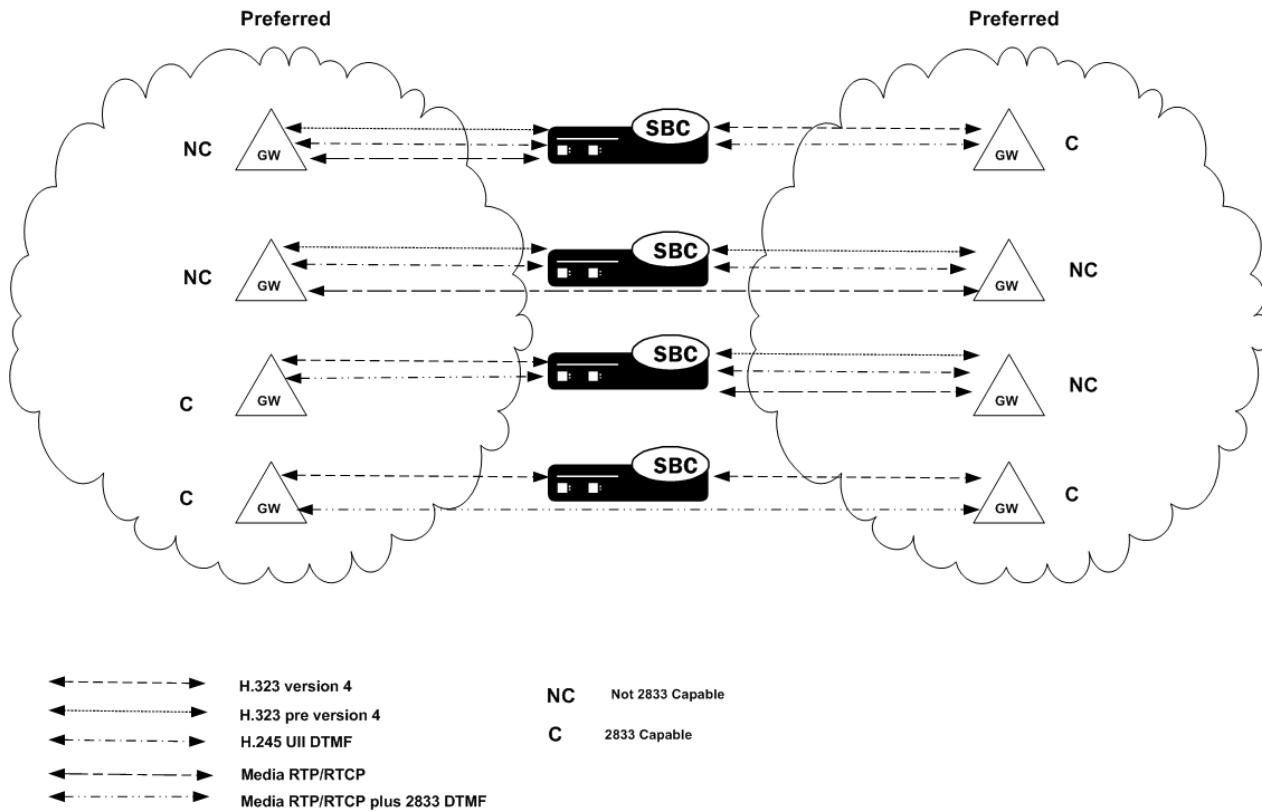
- Preferred: the Net-Net SBC only generates and detects 2833 for endpoints if they negotiate support for 2833 through signaling
- Transparent: the Net-Net SBC offers and answers based on end-to-end signaling and transparently relaying 2833

Preferred 2883 Support

If one side of the call, or a SIP interface, or a session agent, is configured for preferred 2833, the Net-Net SBC only generates and detects 2833 for endpoints if they signal support for 2833. The Net-Net SBC will offer 2833 in the TCS SDP, even if the originating caller did not.

- When the Net-Net SBC manages calls originating from a preferred source going to a preferred target, it:
 - Performs 2833 translation for an endpoint when the originating side requests 2833 but the target does not negotiate 2833
 - Allows 2833 to pass through if the originating side and target of the call are configured as preferred and negotiate 2833
- When the Net-Net SBC manages calls originating from a preferred source going to a transparent target, it:
 - Performs 2833 translation when the originating side requests 2833 but the target is configured as transparent and does not negotiate 2833.

- Allows 2833 to pass through if the originating side and the target of the call are configured as transparent and negotiate 2833. The Net-Net SBC does not perform active translation because both ends support 2833.



If one SIP endpoint does not signal 2833 capability, but the other SIP or H.323 endpoints do, the Net-Net SBC does not perform 2833 translation.

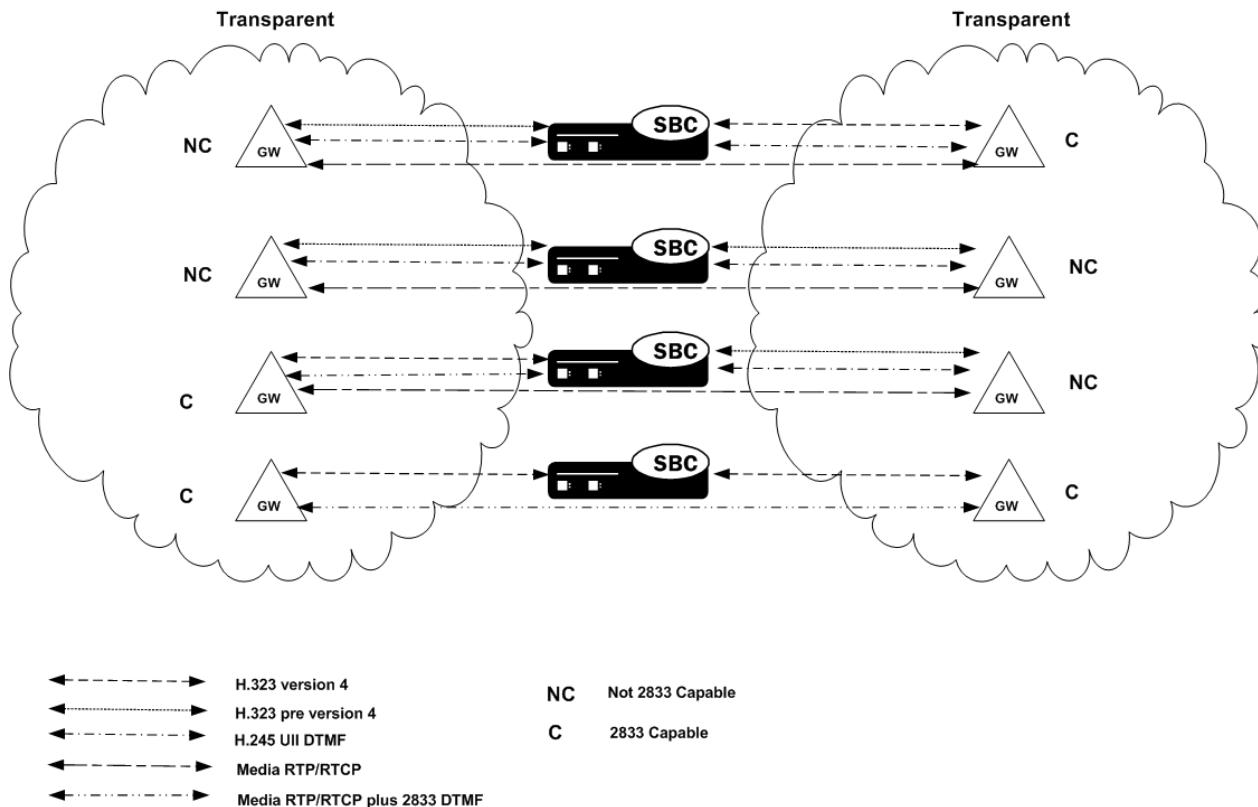
Transparent 2833 Support

The default configuration of the Net-Net SBC for H.323 is transparent 2833. The Net-Net SBC passes on the offered capabilities to the next-hop signaling element. If the next-hop endpoint is for a transparent 2833 target, typical capability negotiation determines the DTMF method. The Net-Net SBC transparently relays the DTMF as it has in previous releases.

With transparent 2833, the Net-Net SBC acts as a typical B2BUA or B2BGW/GK. However when the target of the call is configured as preferred 2833, the Net-Net SBC:

- Relays the 2833 packets if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target signals 2833
- Performs 2833 translation if the originating endpoint does not signal 2833 and the next-hop endpoint for the preferred target does signal 2833

- Does not perform 2833 translation or transparently relay 2833 if the originating endpoint signals 2833 and the next-hop endpoint for the preferred target (or even a transparent 2833 target) does not signal 2833.



Payload Type Handling

The Net-Net SBC supports the RTP NTE for telephony events such as transport of DTMF digits and hook flash. Using RTP NTE, endpoints perform per-call negotiation of the DTMF transfer method and negotiate payload type value for the RTP NTE packets.

Although most endpoints use payload type number 101, the RTP payload type formats can become asymmetrical when being interworked between SIP and H.323 because there is no default standard and endpoints use different types. This means that the payload type negotiated on one side of the Net-Net SBC, and that ends up being used for the call, might not be the same payload type negotiated on the other side of the Net-Net SBC. And while certain endpoints handle the asymmetry well, others do not.

Consider the simplified example of an IWF call initiated in SIP and translated to H.323. In this scenario, the SIP endpoint negotiates the payload type 106 with the Net-Net SBC. And despite the fact that the H.323 endpoint negotiates payload type 101, the Net-Net SBC returns type 106 and the call proceeds using type 106.

However, you can enable forced symmetric payload type handling so the Net-Net SBC changes the payload type of RFC 2833 packets to avoid using asymmetrical payload types.

For H.323 session agents and H.323 interfaces (stacks), you can configure an option that forces symmetric payload type use. The Net-Net SBC can detect when the

payload types negotiated by the SIP and H.323 endpoints are symmetrical and when they are not. When it detects asymmetrical payload type use, the Net-Net SBC forces the remote endpoint to use the RFC 2833 payload type you configure in the SIP interface.

Basic RFC 2833 Negotiation Support

If H.323, SIP, or session agents on either side of the call are configured for preferred 2833 support, the Net-Net SBC supports end-to-end signaled negotiation of DTMF on a call-by-call basis. If the calling party is not configured for preferred support but sends 2833, the Net-Net SBC sends 2833 to the next-hop called party. If the calling party sends H.245 signals or alphanumeric UII, the Net-Net SBC sends H.245 signals or alphanumeric UII to the next-hop called party (if it is an H.323 next-hop).

The Net-Net SBC also supports hop-by-hop negotiation of DTMF capability on a call-by-call basis, if the signaling protocols or session agents on either side of the call are configured for preferred 2833 support.

H.323 to H.323 Negotiation

The Net-Net SBC serves as the H.323 called gateway. It answers RFC 2833 audio telephony event capability in the version 4 H.323/H.245 TCS when it receives a call from an H.323 endpoint configured for preferred RFC 2833.

If the Net-Net SBC is the answering device, configured for preferred support, and the calling device sends 2833, the Net-Net SBC accepts the 2833 regardless of the next-hop's DTMF capabilities. The received dynamic RTP payload type is used for detecting 2833 packets, while the response dynamic payload type is used for generating 2833 packets.

The Net-Net SBC supports:

- RFC-2833 audio telephony events in the version 4 H.323/H.245 TCS as the H.323 calling gateway, when the Net-Net SBC calls an H.323 endpoint configured for preferred RFC 2833 support. The Net-Net SBC sends 2833 to the called party regardless of whether the calling party sends it.
- H.245 UII and RFC-2833 packets sent at the same time, to the same endpoint, even if only half of the call is being provided 2833 support by the Net-Net SBC.

If one half of the call supports H.245 UII, and the other half is being provided 2833 translation by the Net-Net SBC, the Net-Net SBC can also forward the H.245 UII it receives to the 2833 endpoint. For example, when the signaling goes through a gatekeeper or third party call control, sending the H.245 UII in the signaling path allows those devices to learn the DTMF digits pressed.

Signal and Alpha Type Support

The Net-Net SBC supports:

- H.245 signal and alpha type UII in the H.323/H.245 TCS as the H.323 calling gateway when the:
 - Net-Net SBC calls an H.323 endpoint configured for transparent 2833 support
 - calling endpoint's target is configured as preferred

If the originating preferred side also sends 2833, the Net-Net SBC forwards it to the transparent side. The Net-Net SBC sends signal and alpha UII support to the called party regardless of whether the calling party sends it, if the call originates from a preferred side to a transparent side.

- H.245 alphanumeric UII for DTMF for H.323 endpoints that do not signal 2833 or contain explicit H.245 UII capability, for stacks configured for transparent 2833 support.

When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Net-Net SBC translates incoming H.245 UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Net-Net SBC relays the H.245 UII messages.

- H.245 signal type UII for DTMF for H.323 endpoints that do not signal 2833, but do signal explicit H.245 UII capability, for stacks configured for transparent 2833 support.

When the other half of the call is an H.323 endpoint of a stack configured for preferred 2833, the Net-Net SBC translates incoming H.245 signaled UII on the transparent side, to 2833 packets on the preferred side, and vice versa. If the other half of the call is an H.323 endpoint of a transparent stack, the Net-Net SBC relays the H.245 UII messages if both sides support it.

H.323 to SIP Calls

This section explains DTMF interworking specific to H.323 to SIP calls.

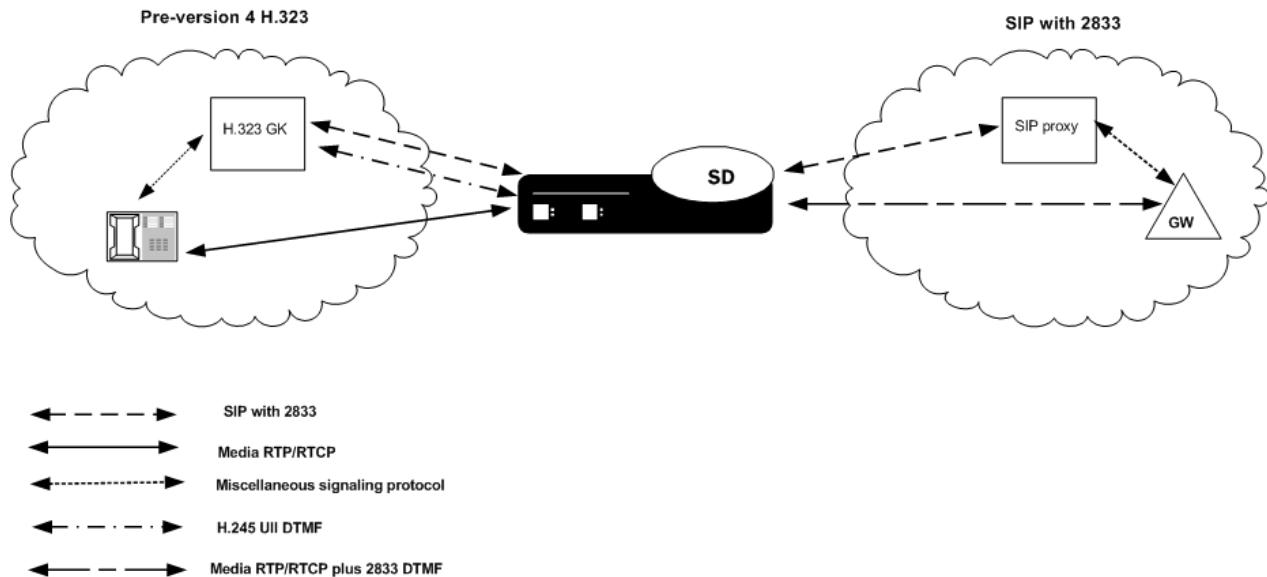
SIP Endpoints

SIP endpoints include those that support:

- RFC 2833
- SIP INFO method

H.323 Non-2833 interworking with SIP

RFC 2833 and the SIP INFO method can be used for conveying DTMF information for SIP based-services. (RFC 2833 is the most widely used.) To provide end-to-end DTMF for SIP devices supporting RFC-2833 interworking with H.323 devices that do not, an RFC 2833 to H.323 UII interworking function is provided.



How H.323 to SIP Calls Work

For H.323 to SIP IWF calls, if 2833-related information is to be sent in the INVITE, the SIP interface of the SIP session agent has to be configured with the **rfc2833-mode** parameter set to **preferred**.

The following example shows an INVITE without 2833 in the SDP:

```

Apr  5 04:28:50.073 On 127.0.0.1:5070 sent to 127.0.0.1:5060
INVITE sipp:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5070;branch=z9hG4bKI WF0000gI 2018604agg71c0;acme_i real m=exte
rnal ; acme_sa=192.168.1.6
Contact: "j doe" <sipp:127.0.0.1:5070>
GenericID: 114421133000000@000825010100
Supported: 100rel ^M
From: "msmith" <sipp:192.168.200.68:5060>;tag=000000ab00011940
To: <sipp:780@192.168.200.6:5060>
Call-ID: 7f00000113ce000000ab000101d0@127.0.0.1
CSeq: 2 INVITE
Content-Length: 225
Content-Type: application/sdp
v=0
o=IWF 3 3 IN IP4 192.168.1.6
s=H323 Call
c=IN IP4 192.168.1.6
t=0 0
m=audio 5214 RTP/AVP 0 18
a=rtpmap:0 PCMU/8000/1
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
m=video 5216 RTP/AVP 31
a=rtpmap:31 H261/9000/1

```

SIP INFO—RFC 2833 Conversion

The Net-Net SBC can perform SIP INFO—RFC 2833 conversion. The Net-Net SBC also provides a way for you to enable a dual conversion mode, where the Net-Net SBC:

- Inserts telephone-event in the SDP offer
- Generates both RFC 2833 event packets and SIP INFO messages regardless of whether or not the SDP offer indicates RFC 2833

You can enable this feature either for SIP interfaces or session agents. The following apply:

- If the next hop SIP interface or session agent's **rfc2833-mode** is set to **preferred**, then the SD inserts RFC 2833 into the SDP offer/answer. This occurs regardless of whether:
 - The original SDP on the opposite side of the call does not support RFC 2833
 - The opposite side's SIP interface or session agent is set to **transparent** mode
- If the next hop SIP interface or session agent is set to **transparent**, then the behavior of the SD depends on the previous hop.
 - If the previous hop is a SIP interface or session agent configured for **transparent** mode, then the SD does not perform any conversion.
 - If the previous hop is a SIP interface or session agent configured for **preferred** mode, the SD does not insert RFC-2833 into the SDP on the

transparent side. It does, however, translate from RFC 2833 to SIP INFO if the originating endpoint supports RFC 2833.

ACLI Instructions and Examples

RFC 2833 Mode for H.323 Stacks

To configure RFC 2833 mode for H.323 stacks:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the media-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.
ACMEPACKET(session-router)# **h323**
4. Type **h323-stacks** and press <Enter>.
ACMEPACKET(h323)# **h323-stacks**
ACMEPACKET(h323-stack)#
- From this point, you can configure H.323 stack parameters. To view all H.323 stack parameters, enter a ? at the system prompt.
5. **rfc2833-mode**—Set the RFC2833 mode. The default value is **transparent**. Valid values are:
 - **transparent**—The Net-Net SBC and H.323 stack behave exactly the same way as before and the 2833 or UII negotiation is transparent to the Net-Net SBC.
 - **preferred**—The H.323 stack uses 2833 for DTMF transfer, which it signals in its TCS. However, the remote H323 endpoint makes the decision. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack reverts back to using UII. You configure the payload format by configuring the h323-config element.

RFC 2833 Payload for H.323

To configure the RFC 2833 payload in preferred mode:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **h323**
From this point, you can configure global H.323 parameters. To view all H.323 configuration parameters, enter a ? at the system prompt.
4. **rfc2833-payload**—Enter a number that indicates the payload type the Net-Net SBC will use for RFC 2833 packets while interworking 2833 and UII. The default value is **101**.
 - Minimum—96

- Maximum—127

Configuring the SIP Interface

You configure the 2833 mode and payload for the SIP interface. You must configure the payload the Net-Net SBC will use for RFC 2833 packets, while interworking 2833 and INFO/UII.

To configure the SIP interface:

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

From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a ? at the system prompt.
4. **rfc2833-payload**—Enter a number that indicates the payload type the SIP interface will use for RFC 2833 packets while interworking 2833 and UII. The default value is **101**.The valid range is:
 - Minimum—96
 - Maximum—127
5. **rfc2833-mode**—Set the RFC 2833 mode for the SIP interface. The default value is **transparent**. Valid values are:
 - **transparent**—The SIP INFO and RFC 2833 translation is transparent to the Net-Net SBC.
 - **preferred**—The RFC 2833 transfer method is the preferred method for sending DTMF, and a telephone event is inserted in the SDP of the outgoing offer. The actual method of transfer, however, depends on the SDP offer/answer exchange that occurs between the Net-Net SBC and remote endpoint. If the remote endpoint supports RFC 2833, the Net-Net SBC performs SIP INFO—RFC 2833 conversion.
 - **dual**—The Net-Net SBC behaves the same as it does when set to **preferred** mode, and it forwards both the original DTMF mechanism and the translated one to the remote endpoint.

Configuring Session Agents

You configure session agents with:

- payload type the Net-Net SBC wants to use for RFC 2833 packets while interworking 2833 and UII.

The default value for this attribute is 0. When this value is zero, the global rfc2833-payload configured in the h323-configuration element will be used instead. For SIP session agents, the payload defined in the SIP interface is used, if the SIP interface is configured with the preferred RFC 2833 mode.
- 2833 mode

A value of transparent or preferred for the session agent's 2833 mode will override any configuration in the h323-stack configuration element.

To configure session agents:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# **session-router**
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **session-agent**
ACMEPACKET(session-agent)#
 4. **rfc2833-mode**—Set the RFC 2833 mode you want the session agent to use. The default value is **none**. Valid values are:
 - **none**—RFC 2833 to UII interworking is based on the H.323 stack configuration.
 - **transparent**—The RFC 2833 or UII negotiation is transparent to the Net-Net SBC. This overrides the H.323 stack configuration, even if the stack is configured for preferred mode.
 - **preferred**—RFC 2833 for DTMF transfer is preferred, which is signaled in the TCS. If the endpoint supports 2833, 2833 is used. If not, the H.323 stack configured as preferred will revert back to using UII. This overrides any configuration in the h323-stack even if the stack is configured for transparent mode.

For SIP INFO—RFC 2833 conversion, you can choose:

- **none**—The 2833-SIP INFO interworking will be decided based on the sip-interface configuration.
- **transparent**—The session agent behaves the same as it did without the SIP INFO—RFC 2833 conversion feature. The SIP INFO and RFC 2833 translation is transparent to the Net-Net SBC.
- **preferred**—The RFC 2833 transfer method is the preferred method for sending DTMF, and a telephone event is inserted in the SDP of the outgoing offer. The actual method of transfer, however, depends on the SDP offer/answer exchange that occurs between the Net-Net SBC and remote endpoint. If the remote endpoint supports RFC 2833, the Net-Net SBC performs SIP INFO—RFC 2833 conversion.
- **dual**—The Net-Net SBC behaves the same as it does when set to **preferred** mode, and it forwards both the original DTMF mechanism and the translated one to the remote endpoint.
5. **rfc2833-payload**—Enter a number that indicates the payload type the session agent will use for RFC 2833 packets while interworking 2833 and UII. The default value is 0. The valid range is:
 - Minimum—0, 96
 - Maximum—127

Enabling Payload Type Handling

You can configure H.323 session agents and H.323 interfaces (stacks) with an option that forces symmetric payload type use. For [Payload Type Handling \(637\)](#) to work properly, you must set the following SIP interface and the global H.323 configuration parameters with these values:

- **rfc2833-mode**—Set this parameter to **preferred**; the default is **transparent**.
- **rfc2833-payload**—Set this parameter to the payload type you want forced for the remote endpoint. Your entry will be between **96** and **127**, with **101** as the default.

To enable forced symmetric payload type handling for an H.323 session agent:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

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

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

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

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

If you want to add this option to a pre-existing H.323 session agent, select the one you want to edit.

4. **options**—Set the options parameter by typing options, a <Space>, the option name **Map2833ForceRemotePT** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(session-agent)# options +Map2833ForceRemotePT
```

If you type options and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

To enable forced symmetric payload type handling for an H.323 interface:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

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

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

3. Type **h323-config** and press <Enter>.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

4. Type **h323-stacks** and press <Enter>.

```
ACMEPACKET(h323-config)# h323-stacks
ACMEPACKET(h323-stack)#
```

5. **options**—Set the options parameter by typing options, a <Space>, the option name **Map2833ForceRemotePT** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(h323-stack)# options +Map2833ForceRemotePT
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

6. Save and activate your configuration.

DTMF Transparency for IWF

In certain vendors' implementations of DTMF during SIP/H.323 IWF, there have been discrepancies between the RFC 2833 and UII/INFO negotiations and what type of messages actually get sent. Instead of correcting these errors on its own end, the Net-Net SBC has perpetuated these inaccuracies.

To ensure that the Net-Net SBC always sends the correctly negotiated protocols, a **media-manager-config** parameter called **translate-non-rfc2833-event** has been created. When **translate-non-rfc2833-event** is enabled, the Net-Net SBC always sends the type of messages that were initially negotiated, regardless of the type of messages it may be receiving.

ACLI Instructions and Examples

To enable DTMF transparency for SIP/H.323 IWF:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.

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

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```
4. **translate-non-rfc2833-event**—To enable this feature, set this parameter to **enabled**. If you do not want to use the feature leave it set to its default behavior, **disabled**.

```
ACMEPACKET(media-manager-config)# translate-non-rfc2833-event enabled
```
5. Save and activate your configuration.

RFC 2833 Packet Sequencing

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.

ACLI Instructions and Examples

To send only the last three end 2833 packets for non-signaled digits events:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **media-profile** and press <Enter>.

```
ACMEPACKET(media-manager)# media-profile
ACMEPACKET(media-manager-profile)#
```
4. **rfc2833-end-pkts-only-for-non-sig**—By default, this parameter is **enabled**—meaning that only the last three end 2833 packets are used for non-

signaled digits events. Set this parameter to disabled if you want the entire start-interim-end RFC 2833 packet sequence for non-signaled digit events

5. Save and activate your configuration.

Enhanced H.245 to 2833 DTMF Interworking

Enhanced H.245 to 2833 and SIP INFO to 2833 DTMF interworking addresses issues experienced where the way the Net-Net SBC timestamps audio RTP packets result in dropped digits and digits with a stutter pattern. These occurrences can cause other network devices to deem the packets unrecoverable (due to jitter), meaning that they will never render the digit.

The Net-Net SBC offers the following:

- Timestamp is based on the current time—The Net-Net SBC can compute the timestamp of the egress 2833 packets using the actual time elapsed in milliseconds since the last RTP packet (rather than incrementing the time by 1 sample). Not only does the Net-Net SBC fill out the timestamp field more accurately, but it also recalculates the checksum.
- End-event 2833 messages default behavior—The Net-Net SBC’s new default behavior is to send three end-event 2833 packets only if the DTMF event is received for:
 - An alphanumeric UII or SIP INFO with no duration
 - A signaled UII or SIP INFO with a duration less than the minimum signal duration (the value you configure using the new media manager configuration **min-signal-duration** option)

For a signaled UII or SIP INFO with a duration greater than the minimum signal duration, the Net-Net SBC behaves as it does in prior releases: It sends the initial event packets, any interim packets (if they exist), and the three end packets.

- Configurable duration for the 2833 event—Without the enhancements being configured, the Net-Net SBC uses a 250 millisecond duration for the 2833 event when it receives an alphanumeric UII or a SIP INFO with no specified duration. The result is that 2833 packets are sent at 50-millisecond intervals until the 250 millisecond time expires; then the three end-event packets are sent.

Now the Net-Net SBC allows you to set the duration of these 2833 events using a new **default-2833-duration** parameter (with a 100 millisecond default) in the media manager configuration. In addition, the Net-Net SBC uses this configured value (instead of the duration sent in the signaling message) when it receives an UII or SIP INFO with a duration less than the minimum signal duration. It checks to make sure that the value for the **default-2833-duration** parameter is greater than the minimum signal duration.

- Configurable minimum signal duration value—Without this configured, the Net-Net SBC accepts and uses the duration it receives in the UII or SIP INFO for the 2833 event. However, you can configure this value using the **min-signal-duration** option in the media manager configuration. If the duration the Net-Net SBC receives is less than the threshold, it uses the value configured in the **default-2833-duration** parameter.

If you do not configure this option, then there is no signaling duration threshold.

Note: Timestamp changes and duration changes only take effect when the 2833 timestamp (**rfc-2833-timestamp**) is enabled in the media manager configuration.

ACLI Instructions and Examples

This section shows you how to configure enhancements for H.245 UII/SIP INFO—2833 DTMF interworking.

To enable the Net-Net SBC to calculate the timestamp based on the current time:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **confi gure terminal**
ACMEPACKET(configure)#
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **medi a-manager**
3. Type **media-profile** and press <Enter>.
ACMEPACKET(medi a-manager)# **medi a-manager**
ACMEPACKET(medi a-manager-confi g)#
4. **rfc-2833-timestamp**—Enable this parameter to use a timestamp value calculated using the actual time elapsed since the last RTP packet. The default is **disabled**. Valid values are:
 - enabled | disabled
5. Save and activate your configuration.

To configure a duration for the 2833 event:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **confi gure terminal**
ACMEPACKET(configure)#
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **medi a-manager**
3. Type **media-profile** and press <Enter>.
ACMEPACKET(medi a-manager)# **medi a-manager**
ACMEPACKET(medi a-manager-confi g)#
4. **default-2833-duration**—Set this parameter to the time value in milliseconds for the Net-Net SBC to use when it receives an alphanumeric UII or a SIP INFO with no specified duration; then the three end-event packets are sent. The default value is 100. The valid range is:
 - Minimum—50
 - Maximum—5000
5. Save and activate your configuration.

Setting the Minimum Signal Duration

To configure the minimum signal duration value:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **confi gure terminal**
ACMEPACKET(configure)#
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **medi a-manager**
3. Type **media-profile** and press <Enter>.
ACMEPACKET(medi a-manager)# **medi a-manager**
ACMEPACKET(medi a-manager-confi g)#

4. **options**—Set the options parameter by typing **options**, a <Space>, the option name **min-signal-duration=x** (where x is the value in milliseconds you want to use for the threshold) with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(mediamanager-config)# options +min-signal-duration=200
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to the configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

SIP Tel URI Support

The Net-Net SBC maps H.323 addresses to either SIP URIs or Tel URIs. You can configure the Net-Net SBC to include Tel URIs in the following SIP headers for calls that require the IWF:

- Request Line
- To
- From

When Tel URI support is not used on a Net-Net SBC performing IWF translations, the SIP INVITE is formatted like it is in the following example. This example uses 192.168.5.5 as the external proxy address, or the next hop (as configured in the local policy).

```
INVITE sip:602@192.168.5.5:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.5:5060;branch=z9hG4bK1WF0aqoqg001g11a7kos4g0
Contact: <sip:603@192.168.5.5:5060>
From: <sip:603@192.168.5.5:5060>;tag=4069ac210018a0
To: <sip:602@192.168.5.5:5060>
```

In the example above, the session needs to be routed to another SIP proxy that can resolve an E.164 number to a SIP address. However, the next SIP proxy must be informed that the message will be routed based on the included E.164 number; the SIP address of the Request URI does not have a routable SIP address. To devise a routable address, the Request URI must be reconstructed as a Tel URI.

Without Tel URI support configured, the terminating SIP user would be required to have an address of 602@192.168.5.5, where the IPv4 address portion is the same as the address for the proxy. If it were not the same, then the session would terminate at the proxy. However, the proxy would be unable to handle the session because the SIP address it received would be unknown/unroutable.

Because it is not desirable to have an IPv4 address be the user-identity and rely on the configuration of the IP network, the SIP INVITE generated by the Net-Net SBC and sent to the proxy must have the following format if it is sent to an H.323 entity.

```
INVITE tel:2345 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.5:5060;branch=z9hG4bK1WFaqoqq00c0bgf9so10o0
Contact: <sip:1234@192.168.5.5:5060>
From: <tel:1234>;tag=4069ac35000c5ff8
To: <tel:2345>
Call-ID: 7f0000113ce4069ac35000c5440
CSeq: 1 INVITE
Content-Length: 155
Content-Type: application/sdp
```

ACLI Instructions and Examples

You enable this feature in the SIP interface configuration.

To configure SIP Tel URI support for calls that require the IWF:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# **session-router**
3. Type **sip-interface** and press <Enter>.
ACMEPACKET(session-router)# **sip-interface**
From this point, you can configure SIP interface parameters. To view see all SIP interface parameters, enter a ? at the system prompt.
4. **teluri-scheme**—Enable or disable the conversion of SIP URIs to Tel URIs. The default value is **disabled**. Valid values are:
 - enabled | disabled
 ACMEPACKET(sip-interface)# **teluri-scheme enabled**
ACMEPACKET(sip-interface)# **done**

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-stacks** and press <Enter>.
ACMEPACKET(h323)# **h323-stacks**
ACMEPACKET(h323-stacks)#
5. Use the ACLI **select** command add this feature to an existing H.323 interface.
ACMEPACKET(h323-stacks)# **select**
6. If you are adding this service to a new H.323 interface, type **option inbandTone** and press <Enter>.

```
ACMEPACKET(h323-stacks)# option inbandTone
```

7. If you are adding this service to an H.323 interface that already exists, type **select** to select the interface to which you want to add the service. Then use the options command and prepend the option with a “plus” (+) sign.
 - If you know the 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 you 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 to 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

Q.850 Cause Value		SIP Status	Comments
16	Normal calling clearing	BYE message	A call clearing BYE message containing cause value 16 normally results in the sending of a SIP BYE or CANCEL request. However, if a SIP response is to be sent to the INVITE request, the default response code should be used.
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from the user	480	Temporarily unavailable
20	Subscriber absent	480	Temporarily unavailable
21	Call rejected	603	Decline (if location field 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

Q.850 Cause Value		SIP Status	Comments
47	Resource unavailable unspecified	503	Service unavailable
55	Incoming calls barred with CUG	403	Forbidden
57	Bearer capability not authorized	403	Forbidden
58	Bearer capability not presently available	503	Service unavailable
65	Bearer capability not implemented	488	Not acceptable here
69	Requested facility not implemented	501	Not implemented
70	Only restricted digital information available	488	Not acceptable here
79	Service or option not implemented, unspecified	501	Not implemented
87	User not member of CUG	403	Forbidden
88	Incompatible destination	503	Service unavailable
102	Recovery on timer expiry	504	Server time-out

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:
 - Minimum—100

- Maximum—699
6. **q850-cause**—Set the Q.850 cause code that you want to map to the SIP response code that you set in step 5. There is no default.
 7. **q850-reason**—Set the Q.850 reason corresponding to the Q.850 cause code that you set in step 6. There is no default. If your value has spaces between characters, then your entry must be surrounded by quotation marks.
 8. Repeat this process to create the number of local response map entries that you need.
 9. Save and activate your configuration for changes to take effect.

To configure a Q.850 cause to a SIP status with reason mapping:

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
 - Minimum—100
 - Maximum—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**—Enable this parameter to add the Reason header to IWF calls. The default is **disabled**. Valid values are:

- enabled | disabled

IWF Privacy: Caller Privacy on Unsecure Networks

This feature enables bi-directional SIP/H.323 IWF support for CPID hiding by using the presentation indicators in the Calling Party Number information element for H.323 signaling, and RFC 3325-based privacy support for SIP signaling. It lets the 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 'j doe\000'
Information element: Display
Length: 9
Display information: j doe\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 si p: 780@192.168.200.6:5060; acme_real m=internal SIP/2.0
Via: SIP/2.0/UDP
127.0.0.1:5070; branch=z9hG4bK1WF00000510d031s9kou5c0; acme_i real m=external
Contact: "Anonymous" <si p: anonymous@127.0.0.1:5070
GenericID: 7400000@000825010100
Supported: 100rel
From: "Anonymous" <si p: anonymous@anonymous.ivalid>; tag=0000004a000d8cc0
To: <si p: 780@192.168.200.6:5060
Call-ID: 7f00000113ce0000004a000d88d8@127.0.0.1
CSeq: 2 INVITE
P-Asserted-Identity: "j doe" <si p: 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]192.168.200.6:5060 received from
192.168.200.6:5062
INVITE sip:800@192.168.200.6:5060 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@192.168.200.6:5060
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_i=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@192.168.200.6:5060
```

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 'j doe\000'
Information element: Display
Length: 9
Display information: j doe\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: sip <sip:7789@192.168.200.6:5062>;tag=1
To: sut <sip:800@192.168.200.68:5060>
P-Asserted-Identity: sip <sip:7789@192.168.200.6:5062>
Call-ID: 1.1336.192.168.200.6@sip.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_i=internal SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK000c0210385hv9gpt001.1
From: sip <sip:7789@192.168.200.6:5062>;tag=SDk0j_pc01-1
To: sut <sip:800@192.168.200.68:5060>
Call-ID: SDk0j_pc01-8e15e11e7f9a20523462972843c7e579-06ahc21
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 'sip'
Information element: Display
Length: 4
Display information: sip
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

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: j doe\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 sipp:780@192.168.200.6:5060 SIP/2.0
Via: SIP/2.0/UDP
192.168.200.68:5060;branch=z9hG4bK6810pr20205h2akqe381.1
Contact: "Anonymous" <sipp:anonymous@192.168.200.68:5060;transport=udp>
Supported: 100rel
From: "Anonymous" <sipp:anonymous@anonymous.inval.id>;tag=SDfd9sa01-
000000ba00023280
To: <sipp:780@192.168.200.6:5060>
Call-ID: SDfd9sa01-6f93292521b83a0980647f34451c5afed-06ahc21
CSeq: 2 INVITE
P-Preferred-Identity: "rdoe" <sipp: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:

- In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
- Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#
- 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      real m=private
2: 124.21.5.3       real m=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(real m-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 \(655\)](#) and [IWF Privacy: Caller Privacy on Secure Connections \(659\)](#), 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 **1D**. 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 'j doe\000'
    Information element: Display
Length: 9
    Display information: j doe\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: "j doe"<sip:42343@192.168.200.68:5060;transport=udp>
Supported: 100rel
From: "j doe"<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@192.168.200.6:5060 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.calld.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 'sip'
Information element: Display
Length: 4
Display information: sip
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-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 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.

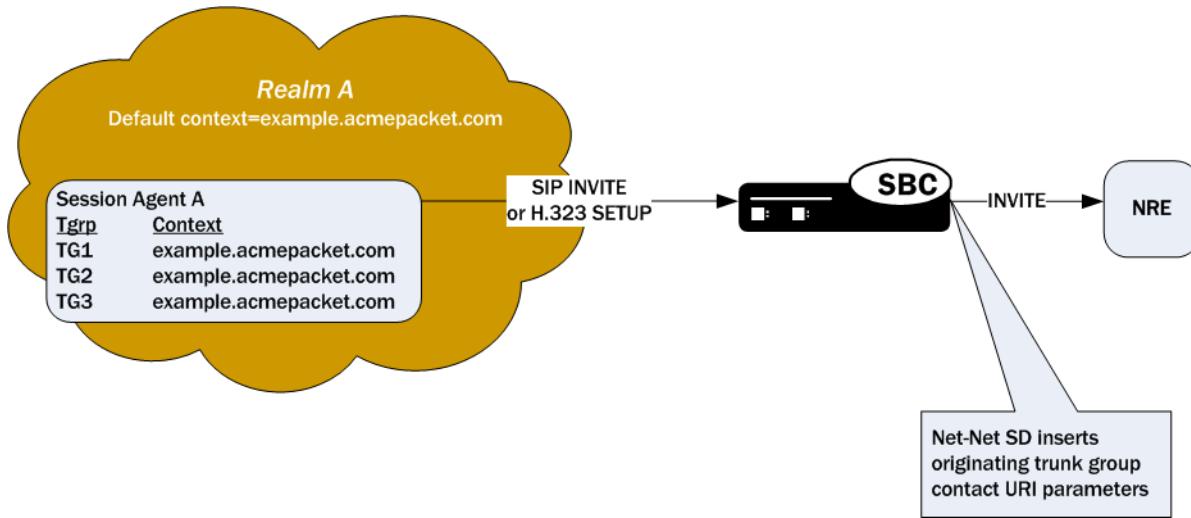
In the absence of official SIP standards for transporting trunk groups between signaling elements, the Net-Net SBC allows you to define URI parameters for use with originating and terminating trunk group URIs.

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 trunk-context = tel co. example. com:

tel : +15555551212; tgrp=tg55; trunk-context=tel co. example. com

2. The second format is customized specifically for access URIs and contains two provisioned parameters: tgrp (or tgname) and context (or provstring). This appears as tgrp.context, where these definitions apply:

- tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
- context (provstring)—Name of the originating trunk group context; this value must have at least one alphabetical character in the top label

This format conforms to format for a hostname in the SIP URI as specified in RFC 3261, such that a trunk group identifier would appear as:

custsi te2NY-00020. type2. voi p. carrier. net

where the tgrp is custsite2NY-00020, and the context is type2.voi.p.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

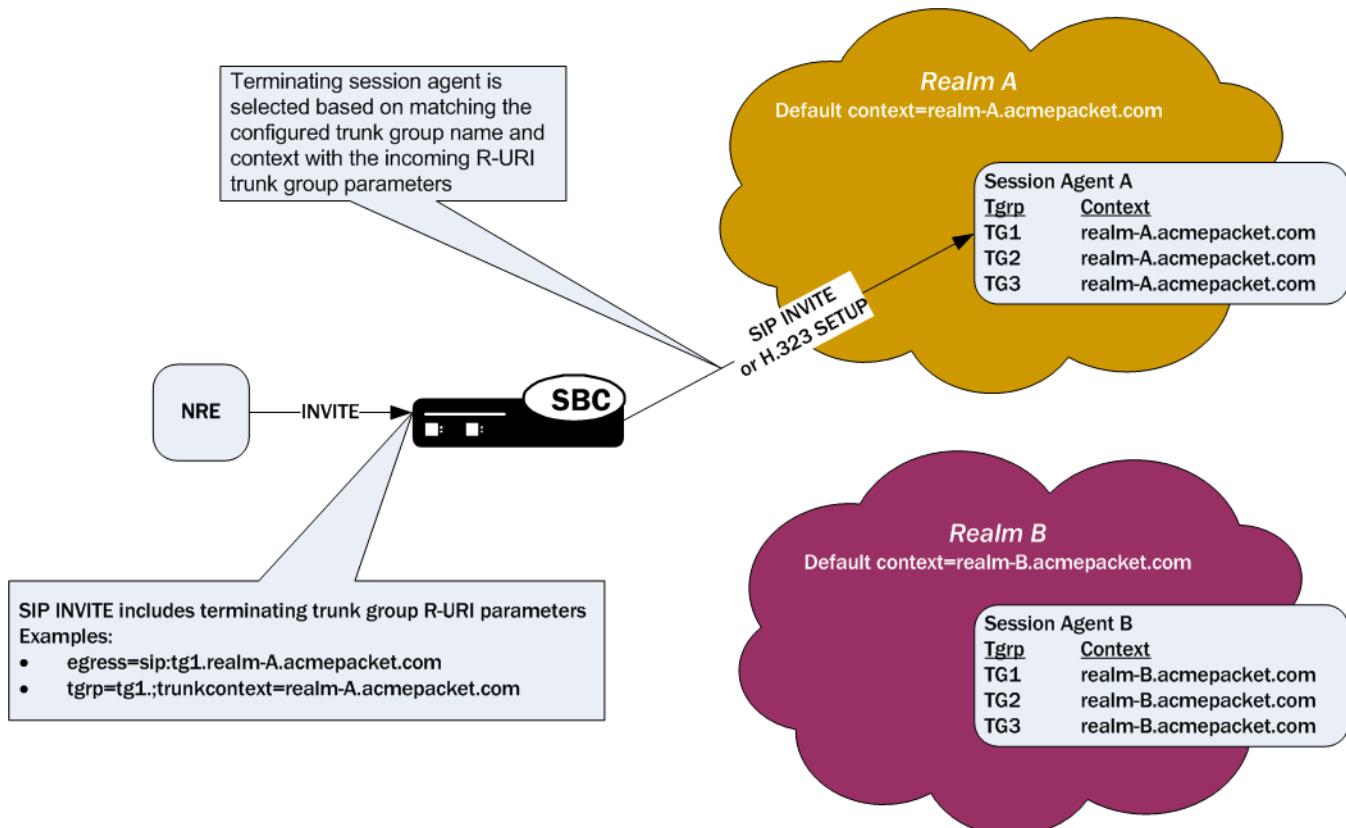
other-param = accessid / pname [ '=' pvalue ]
accessid = "access=" accessURI

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 i sp. exempl e. net@egwy. i sp. exempl e. net:

```
I INVITE sip:+15555551212; tgrp=TG2-1; trunk-
context=i sp. exempl e. net@egwy. i sp. exempl e. net SIP/2.0
```

2. The second format is customized specifically for egress URIs and contains two provisioned parameters: tgrp (or tgname) and context (or tgdomain). This appears as tgrp.context (or tgname.tgdomain), where definitions apply:

- tgrp (tgname)—Provisioned trunk group name for the originating session agent; this value must have at least one alphabetical character, cannot contain a period (.), and can contain a hyphen (-) but not as the first or the last character
- context (tgdomain)—Name of the terminating trunk group context; this value can be up to twenty-four characters

The use of multiple terminating trunk groups is not supported.

The BNF for a single, egress URI with trunk group information conforms to:

```
SIP-URI = "sip: " [userinfo] hostport uri-parameters [headers]
uri-parameters = *( ";" uri-parameter )
uri-parameter = transport-param / user-param / method-param
/ ttl-param / maddr-param / lr-param / other-param

other-param = egressid / pname [ '=' pvalue ]
egressid = "egress=" egressURI
egressURI = scheme tgname ["."] tgdomain
scheme = "sip:" / token
tgname = ALPHA / *(alphanum) ALPHA *(alphanum / "-") alphanum /
alphanum *(alphanum / "-") ALPHA *(alphanum) # up to 23 characters
tgdomain = *(domain ".") toplabel # up to 24 characters
toplabel = ALPHA / ALPHA *(alphanum / "-") alphanum
domain = alphanum/ alphanum *(alphanum / "-") alphanum
```

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 for routing for terminating trunk group URIs. The default value is **none**. Valid values are:
 - **none**—Disables routing based on trunk groups
 - **iptel**—Uses trunk group URI routing based on the IPTEL formats

- **egress-uri**—Uses trunk group URI routing based on the egress URI format

Configuring a Session Agent for Trunk Group URIs

In a session agent, you can configure the outbound or inbound SIP header manipulation rules to use, as well as a list of trunk group names and contexts. For the trunk group names and contexts, you can use either the IPTEL or the custom format.

To configure a session agent for trunk group URIs:

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# configure terminal
2. Type **session-router** and press <Enter> to access the session-related configurations.
ACMEPACKET(configure)# session-router
3. Type **session-agent** and press <Enter>
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
4. **out-manipulationid**—Enter the name of the SIP header manipulations configuration that you want to apply to the traffic exiting the 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.

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(real-m-config)#{}
4. **trunk-context**—Enter the trunk group context to use for this realm. There is no default.

Using this Feature with SIP Interface Registration Caching

If you are using the trunk group URIs feature with SIP interface that has registration caching enabled, then you need to configure the **preserve-user-info** option for that SIP interface.

1. In Superuser mode, type **configure terminal** and press <Enter>.
- ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the session-related configurations.
- ACMEPACKET(configure)# **session-router**
3. Type **session-group** and press <Enter>.
- ACMEPACKET(session-router)# **sip-interface**

ACMEPACKET(sip-interface) #

4. **options**—Add support for trunk group URIs with SIP interface that uses registration caching.

If you are adding this option to a new SIP interface, you can just type **options**, a <Space>, and **preserve-user-info**.

If are adding this to an existing SIP interface, you must type a “plus” (+) sign before the option or you will remove any previously configured options. In order to append the new option to the options list, you must prepend the new option with a “plus” sign: **options +preserve-user-info**.

Example 1: Adding Originating Trunk Group Parameters in IPTEL Format

This ACLI sample shows you how the ACLI SIP manipulations might appear in a case where you want to add originating trunk parameters in IPTEL format.

```

sip-manipulation      add_ipTEL
header-rule
  name          contact
  action        manipulate
  match-value
  msg-type      any
element-rule
  name          tgrp
  type          uri-user-param
  action        add
  match-val-type
  match-value
  new-value     $TRUNK_GROUP
element-rule
  name          trunk-context
  type          uri-user-param
  action        add
  match-val-type
  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      add_att
header-rule
  name          contact
  action        manipulate
  match-value
  msg-type      any
element-rule
  name          egressURI
  type          uri-param
  action        add
  match-val-type
  match-value
  new-value     $TRUNK_GROUP
"sip: "+$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          strip_ipTEL
  name                      request-uri
  header-rule
    name                    manipulate
    action
    match-value
    msg-type
    element-rule
      name                  tgrp
      type
      action
      match-val-type
      match-value
      new-value
  element-rule
    name
    type
    action
    match-val-type
    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          strip_egress
  name                      request-uri
  header-rule
    name                    manipulate
    action
    match-value
    msg-type
    element-rule
      name                  egressURI
      type
      action
      match-val-type
      match-value
      new-value

```

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

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 how 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-UIIE 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 PAI, 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 **iwf-config** and press <Enter>. The system prompt will change to let you know that you can configure individual

ACMEPACKET(session-router)# **iwf-config**
4. **options**—Set the options parameter by typing **options**, a <Space>, the option names with a “plus” sign in front, and then press <Enter>.

Your entry for the **colp-colr-country-code** option require that you type in the entire option name, an equal sign (=), and then the country code value.

To enter both options at once, separate the two with one command and enclose your entire entry in quotation marks (" "); see the following example for command-line syntax.

ACMEPACKET(iwf-config)# **options "+colp-colr-iwf, colp-colr-country-code=1"**

If you type this enter without the “plus” sign, you will overwrite any previously configured options. In order to append options to the IWF configuration’s options list, you must prepend the new options with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

Options for Calls that Require the IWF

You can configure several specific behaviors by configuring options for calls that require the IWF, and set them for the H.323 side of the call. These options are listed and defined in the table below. Options can be configured either globally for the H.323 configuration, individually for an H.323 interface, or for H.323 session agents.

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(sessi on-router)# **h323**

4. Type **h323-stacks** and press <Enter>. The system prompt changes again to let you know that you can begin configuring individual parameters.

ACMEPACKET(h323)# **h323-stacks**

ACMEPACKET(h323-stack)#

From this point, you can configure H.323 interface parameters. To view see all H.323 interface parameters, enter a ? at the system prompt.

5. Type **options**, a space, and the name of the option you want to use. In this example, the MapG729 will map H.245 G.729 to SDP G.729 with Annex B and vice versa.

ACMEPACKET(h323-stack)# **options 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.

Options	Description
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 2000K 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# **config terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
3. Type **h323** and press <Enter>.
ACMEPACKET(session-router)# **h323**
ACMEPACKET(h323)#
4. Type **h323-stacks** and press <Enter>.
ACMEPACKET(h323)# **h323-stacks**
ACMEPACKET(h323-stack)#
- If you are adding support for this feature to a pre-existing H.323 interface (stack), then you must select (using the ACLI **select** command) the configuration that you want to edit.
5. **options**—Set the options parameter by typing **options**, a <Space>, the option name **suppress100rel** with a “plus” sign in front of it, and then press <Enter>.
ACMEPACKET(h323-stack)# **options +suppress100rel**

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new option to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

IWF Codec Negotiation: H.323 Slow Start to SIP

For instances when the 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **iwf-config** and press <Enter>.

```
ACMEPACKET(session-router)# iwf-config
ACMEPACKET(iwf-config)#
```
4. **no-sdp-in-invite**—Enable this parameter 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. Valid values are:
 - enabled | disabled
5. Save and activate your configuration.

IWF: H.245 Signaling Support for G.726

In addition to providing G.726 support for pure SIP and pure H.323 calls, the 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# session-router
```
3. Type **media-profile** and press <Enter>.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```
4. **name**—Set the name of the media profile to G726-16. Values to support this feature are: G726-16, G726-24, G726-32, and G726-40.
5. **media-type**—Set the media type to use for this media profile; for generic video, set this parameter to **audio**. Valid values are:
 - audio | video | application | data
6. **payload-type**—Set the payload type to use for the generic video media profile.
7. **transport**—Set the transport type to use for the generic video media profile. The default value is **RTP/AVP**. Valid values are:
 - UDP | RTP/AVP
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a media profile configuration for H.245/IWF G.726 support:

```
media-profile
  name          g726-40
  media-type    audio
```

payload-type	105
transport	RTP/AVP
req-bandwidth	0
frames-per-packet	0
parameters	
average-rate-limit	0
sdp-rate-limit-headroom	0
sdp-bandwidth	disabled

To set a media profile for generic audio support:

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

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

```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(configure)# session-router

```
3. Type **media-profile** and press <Enter>.

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

```
4. **name**—Set the name of the generic audio media profile to **genericAudioCapability**. There is no default for this parameter.
5. **media-type**—Set the media type to use for this media profile; for generic video, set this parameter to **audio**. Valid values are:
 - audio | video | application | data
6. **payload-type**—Set the payload type to use for the generic audio media profile.
7. **transport**—Set the transport type to use for the generic audio media profile. The default value is **RTP/AVP**. Valid values are:
 - UDP | RTP/AVP
8. Complete the rest of the media profile configuration as needed.
9. Save and activate your configuration.

The following is a sample of a generic audio media profile configuration:

media-profile	
name	genericAudioCapability
media-type	audio
payload-type	104
transport	RTP/AVP
req-bandwidth	0
frames-per-packet	0
parameters	
average-rate-limit	0
sdp-rate-limit-headroom	0
sdp-bandwidth	disabled

Customized G.729 Support

The Net-Net SBC supports the use of custom G.729 encoding for calls that require interworking between SIP and H.323. If you use a proprietary G.729 encoding format in your network, then you might need to use this feature.

When you set the **acceptG729abFormat** option in the global H.323 configuration, the Net-Net SBC performs conversions like those in the following examples:

- For calls initiated in SIP, the Net-Net SBC can parse RTP map strings such as G.729a and G.729ab in the SDP, and then map them to H.245 data types.
 - G.729a becomes g729AnnexA.
 - G.729ab becomes g729AnnexAwAnnexB.
- For calls initiated in H.323, the Net-Net SBC can create non-standard RTP map strings such as G.729a and G.729ab from mapped H.245 data types.
 - g729 becomes G729.
 - g729AnnexA becomes G.729a.
 - g729AnnexAwAnnexB becomes G.729ab.

How It Works

When you enable the **acceptG729abFormat** option, the Net-Net SBC performs customized G.729 mapping in the following instances.

- For calls initiated in SIP and translated to H.323, the Net-Net SBC:
 - Converts the SDP in an incoming SIP INVITE to a list of fastStart OpenLogicalChannel requests that are in turn included in the outgoing Setup message.
 - Converts the list of fastStart OpenLogicalChannelAck responses (which can be received in any message up to and including the Connect message) to SDP sent with a SIP response.
- For calls initiated in H.323 and translated to SIP, the Net-Net SBC:
 - Converts the list of fastStart OpenLogicalChannel requests to SDP in the outgoing SIP INVITE.
 - Converts SDP in a SIP response (such as a 200 OK) to the list of fastStart OpenLogicalChannelAck responses included with the callProceeding, Progress, Alerting, or Connect message. This depends on when the SDP is received on the SIP side.
- For all IWF calls regardless of initiating protocol, the Net-Net SBC:
 - Converts SDP on the SIP side to the terminalCapabilitySet message to be sent on the H.323 side.

Also note that when the format is G729, the Net-Net SBC maps it to g729wAnnexB if the a=fmtp:18 annexb=yes attribute is present. When the a=fmtp:18 annexb=no attribute is present, the Net-Net SBC maps G729 to g729. And with no a=fmtp:18 annexb=no attribute, the Net-Net SBC also maps G729 to g729 when this option is enabled.

The Net-Net SBC also maps G729 to g729 because pure G729 with static payload type 18 does not include an fmtp attribute where annexb=no.

About Dynamic Payload Mapping

G.729a and G.729ab use dynamic payload types, but the Net-Net SBC does not propagate these dynamic payload types to corresponding dynamic RTPPayloadType (an optional field in OpenLogicalChannel requests) on the H.323 side.

For an IWF call initiated in H.323, the dynamic payload types for G.729a and G.729ab are retrieved from media profile configurations when the Net-Net SBC converts the list of fastStart OpenLogicalChannel requests to SDP sent on the SIP side. As a result, you must set up media profile configurations for G.729a and G.729ab for the feature to work properly. In these media profiles, the following parameters must be set as follows:

- **name**—For the G.729a profile, set the **name** to **G.729a**. For the **G.729ab** profile, set the **name** to **G.729ab**.
- **payload-type**—For each media profile (**G.729a** and **G.729ab**), DO NOT use payload type 18, which is the static payload type used for G729.

ACLI Instructions and Examples

This section shows you how to configure the **acceptG729abFormat** option in the global H.323 configuration.

To enable customized G.729 support for IWF calls:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **h323-config** and press <Enter>.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323-config)#
```

If you are adding this feature to a pre-existing configuration, select the configuration to edit it.
4. **options**—Set the options parameter by typing options, a <Space>, the option name **acceptG729abFormat** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(h323-stack)# options +acceptG729abFormat
```

If you type **options** and then the option value for either of these entries without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save and activate your configuration.

International Peering with IWF and H.323 Calls

When you do not enable this feature, SIP to H.323 IWF calls default to a National Q.931 Number Type and it is not possible to change it to an International number. This feature allows you to override that behavior by configuring the option **cpnType=X**, where X is an integer that maps to various Q.931 Number Types. When this option is set, Q.931 Number Type for both calling party and called party are updated to the configured value for all outgoing calls on the h323-stack.

The following is a list of possible `cpnType=X` option values for X:

- 0—Unknown public number
- 1—International public number
- 2—National public number
- 3—Specific public network number
- 4—Public subscriber number
- 5—Public abbreviated number
- 6—Private abbreviated number

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type `h323-config` and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# h323-config
ACMEPACKET(h323)#
```
4. Type `h323-stacks` and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(h323)# h323-stack
ACMEPACKET(h323-stack)#
```
5. Set the options parameter by typing `options`, a <Space>, the option name `cptnType=x` with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(h323-stack)# options +cptnType=x
```

If you type `options` without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the `h323-stack`'s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
6. Save and activate your configuration.

IWF Codec Renegotiation for Audio Sessions

For calls requiring interworking between SIP and H.323, there can be several instances for audio sessions when a mid-call codec change is necessary. These are some examples of when the codec used for voice transportation is necessary:

- Sessions between analog FAX machines that start as regular voice calls but then must use a codec that is fax-signalling tolerant (like transparent G.711) when FAX tones are detected; detection takes place after the call has been answered. The case of modem calls is similar.
- An established call is redirected in one carrier's network either to a different enduser or to a media server. In this case, the party to which the call is redirected might not support the codec used in the redirection. If request for a codec

change is carried out at the signalling level, the call can proceed with the party to which the call was redirected.

- Endusers might want to change codecs when they suffer low voice quality.

Both SIP and H.323 provide mechanisms for changing codecs during a call: SIP uses the ReINVITE, and H.323 uses the H.245 Request Mode. Using the option called `processRequestModeForIWF=all` either in an H.323 interface (stack) or an H.323 session agent configuration, you can enable the Net-Net SBC to interwork SIP ReINVITE and H.245 Request Mode requests.

RTN 1976

Codec Request Change from the SIP Side

When a SIP party requests a code change, the Net-Net SBC communicates with the H.323 endpoint to renegotiate support for an updated codec. In this renegotiation, the Net-Net SBC presents codec for use ordered according to the SIP side's preference and one is selected. Then the Net-Net SBC handles opening of a new logical channel that uses the updated codec, and closes the old logical channel (that uses the now-outdated codec). On the SIP side, the Net-Net SBC sends a 200 OK with the necessary RTP port and codec information for the new logical channel.

Codec Request Change from the H.323 Side

When the Net-Net SBC receives a codec request change on the H.323 side of an IWF call, it sends a Re-INVITE to the SIP endpoint containing new codec and information. The Net-Net SBC uses IP address and port information it has cached for the H.323 side of the call for the Re-INVITE since H.245 Request Mode requests do not have this data. If the IP address and port combination should subsequently change (in an OLC from the H.323 side), the Net-Net SBC handles additional INTVITES on the SIP side to support the change.

Exceptional Cases

When the relevant option is enabled, the Net-Net SBC can handle properly the following cases of codec change:

- When the H.323 side rejects the request mode change, the Net-Net SBC response to the SIP side with a 488 Not Acceptable. Session description and state remain unchanged, and the call continues using the original session description.
- When the H.323 side does not respond to the request mode change within the timeout limitation, the Net-Net SBC releases the call on both sides.
- When the SIP side does not respond to the ReINVITE within in the timeout limitation, the Net-Net SBC releases the call on both sides.
- When the intersection of codec is empty, the Net-Net SBC rejects the codec change on the SIP side with a 488 Not Acceptable and on the H.323 side with an H.245 RequestModeReject. Session description and state remain unchanged, and the call continues using the original session description.
- If the Net-Net SBC does not receive any of the LogicalChannel request or acknowledgement messages, the Net-Net SBC releases the call on both sides.

Note that for protocol timeout errors, the preferred behavior is to release the call on both sides. Timeout errors usually indicate network problems, such as an endpoint being unreachable.

ACLI Instructions and Examples

You can apply the `processRequestMethodForIWF=all` to H.323 interfaces (stacks) and to H.323 session agents (sessions agents for which H.323 has been identified in the `protocol` parameter). The example below shows you how to enable this option for an H.323 session agent.

To enable IWF codec renegotiation for an H.323 session agent:

1. In Superuser mode, type `configure terminal` and press <Enter>.
ACMEPACKET# **config**ure **termi**nal
ACMEPACKET(configure)#
2. Type `session-router` and press <Enter>.
ACMEPACKET(configure)# **sessi**on-**router**
ACMEPACKET(session-router)#
3. Type `session-agent` and press <Enter>. If you are adding this feature to a pre-existing configuration, you will need to select and edit it.
ACMEPACKET(session-router)# **sessi**on-**agent**
ACMEPACKET(session-agent)#
4. **options**—Set the options parameter by typing options, a <Space>, the option name `processRequestMethodForIWF=all` with a “plus” sign in front of it, and then press <Enter>.
ACMEPACKET(session-agent)# **opti**ons +**processRequestMethodForIWF=all**
If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.
5. Save your work.

Introduction

This chapter explains how to configure the Net-Net SBC to support Media Gateway Control Protocol (MGCP/NCS) signaling services.

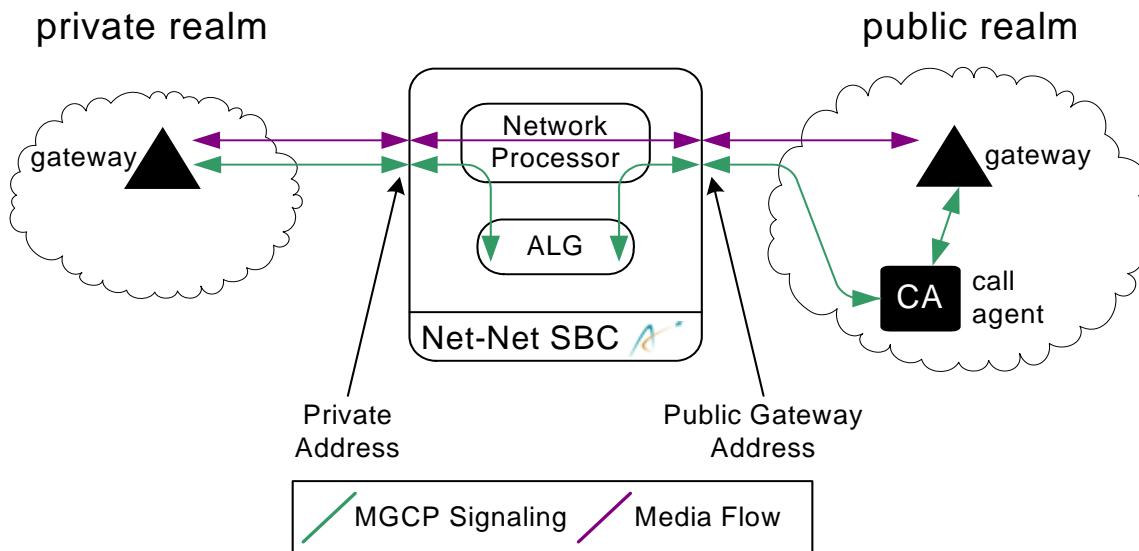
MGCP/NCS Overview

The Net-Net SBC provides MGCP/NCS Application Layer Gateway (ALG) functionality for MGCP/NCS messages between media gateways and media gateway controllers. For the purpose of this document, there are two major types of elements in an MGCP/NCS deployment, endpoints and call controllers. Endpoints encompass Integrated Access Devices (IAD), Multimedia Terminal Adapters (MTA), and Gateways (GW). Call controllers encompass Media Gateway Controllers (MGC), Softswitches, and Call Agents (CA). Throughout this chapter, GW and CA will be used to refer to their respective class of devices. The Net-Net SBC is positioned in the path of packets flowing between media gateways and media gateway controllers to provide a level of packet translation, without directly participating in the media flow.

GWs and CAs are not aware of the role of the Net-Net SBC; they assume a direct connection to each other. To CAs, the Net-Net SBC appears to be a GW. To GWs, the Net-Net SBC appears to be a CA. The Net-Net SBC, positioned as an intermediary device, provides seamless NAT and packet forwarding functionality.

When MGCP/NCS signaling messages traveling between CAs and GWs pass through the Net-Net SBC, the Net-Net SBC identifies the packets as addressed to CAs or GWs. The network processor, in turn, sends these packets to the host processor, which performs the MGCP/NCS NAT and returns the packet to the network processor.

The network processor sends the packet on to its next destination. The following diagram shows an abstract representation of an MGCP/NCS network configuration.



MGCP/NCS and Realms

You need to understand how realms are configured for your MGCP/NCS deployment. A private realm in an MGCP/NCS configuration is the access network. This realm is commonly the public Internet where individual GWs exist. The private realm can also be a VPN. The public realm refers to the service provider's backbone network.

MGCP/NCS configurations must be unique on a per-private realm basis. No two MGCP/NCS configuration elements can have the same private realm. Depending on the design of your network: all of MGCP/NCS configuration elements can share the same public realm; some MGCP/NCS configuration elements can share the same public realm and the remainder can have different public realms; all MGCP/NCS configuration elements can have different public realms. For configurations that include multiple public realms, no two of these public realms can include the same public gateway address and public realm pair.

Note: Public and private realms must be configured first in the realm configuration element, as explained in the Realm Configuration section of this guide.

MGCP/NCS NAT Traversal

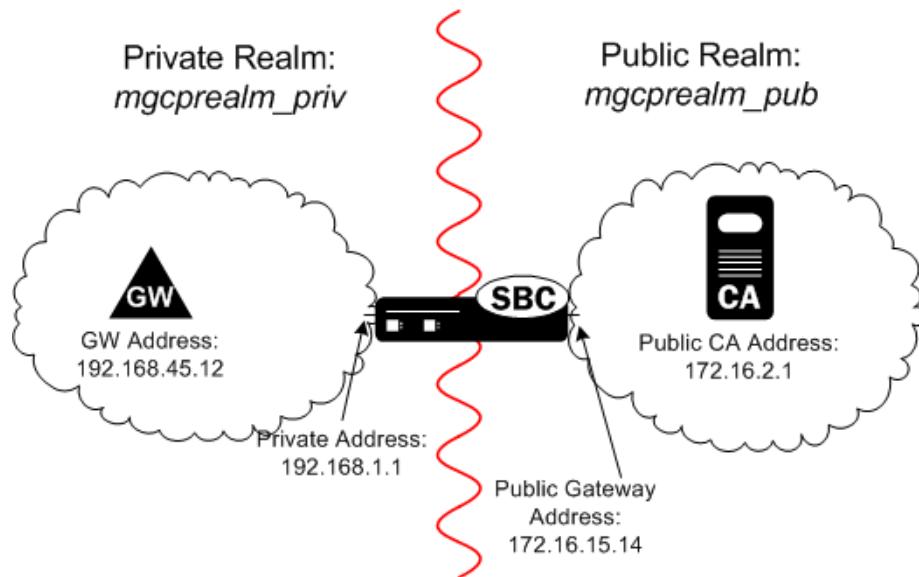
Net-Net SBCs work seamlessly when GWs in the private realm are located behind NATs. The MGCP/NCS configuration provides a mechanism for NAT traversal. When the Net-Net SBC recognizes that a layer 3 IPv4 source address does not match the same information provided in the layer 5 MGCP signaling message, NAT traversal is enabled. The Net-Net SBC sends packets through a NAT at a given interval to keep a pinhole open and prevent MGCP/NCS connections from prematurely closing.

The Net-Net SBC recognizes when an endpoint is behind a NAT because of the discrepancy between the host portion of the endpoint ID and the layer 3 source address. This recognition enables the mechanisms for maintaining NAT traversal. You do not have to explicitly enable NAT traversal unless your deployment uses

FQDNs in the endpoint ID when behind a NAT. In that case, NAT traversal is not automatically enabled; you must explicitly enable it in this situation.

MGCP/NCS Network Topology

The following figure is a Network Topology map used for this section. The example shows three network elements: GW, Net-Net SBC, CA. There are two logical realms: `mgcprealm_priv` and `mgcprealm_pub`. The GW exists in `mgcprealm_priv` and the CA exists in `mgcprealm_pub`. The GW and CA each have one assigned IPv4 address. The Net-Net SBC has two IPv4 addresses, one on each network interface existing in each of the two realms.



MGCP/NCS Configuration Overview

To create a basic MGCP/NCS configuration, populate the MGCP configuration element as follows.

1. Set the private realm information. This includes the private realm ID, and the IPv4 address and port number of the network interface on which the private realm exists. The respective parameters for these fields are **Private Realm**, **Private Address**, and **Private port**. The private address and port pair compose a virtual CA on this Net-Net SBC from the GWs' points of view.
2. Set the public CA information. This includes the IPv4 address or hostname and port number of the call agent located in the public realm. The respective parameters for these fields are **Public Call Agent Host**, **Public Call Agent Port**, **Public Call Agent Address**.
3. Set the public gateway information. This includes the public realm ID, and the IPv4 address (or hostname) and port number of the network interface on which the public realm exists. In addition, a second public gateway port number can be identified. The public realm is where the CAs and DNS servers exist. The public address and port pair compose a virtual GW on this Net-Net SBC from the CA's point of view. The respective parameters for these fields are **Public Realm**, **Public Gateway Host**, **Public Gateway Address**, and **Public Gateway Port**. **A Second Public Gateway Port** can also be configured.
4. Set the NAT traversal configuration. MGCP/NCS NAT must be explicitly enabled when using FQDNs for endpoint IDs. The parameters for this field are

NAT Traversal and Audit Interval. These configuration elements only refer to enabling NAT traversal for GWs that exist in the private realm.

5. Set the ALG port. This field defines the port associated with this MGCP Configuration element. Each MGCP element must have a unique ALG port so that the ALG process can distinguish which MGCP element (stack) is communicating with the network processor. The parameter defining this field is **ALG Port**.

Before You Configure

In order for the Net-Net SBC to pass media and control traffic for MGCP/NCS from the public realm to the private realm and vice versa, you must set the following elements for a baseline configuration:

- physical interfaces
- network interfaces
- media manager
- steering pools
- realm configurations

ACLI Instructions and Examples

This section describes how to configure the basic MGCP/NCS functionality.

To configure general MGCP/NCS information:

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 **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **mgcp-config**
ACMEPACKET(mgcp-config)#[/b]

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

The following example comes from the MGCP configuration scenario described in this chapter's "Network Topology" section. Parameters not described in this section are omitted below.

private-real m	mgcprealm-pri v
private-address	192.168.1.1
private-port	2727
public-real m	mgcprealm-pub
public-ca-host	
public-ca-address	172.16.2.1
public-ca-port	2727
public-gw-host	
public-gw-address	172.16.15.14
public-gw-port	2427
second-public-gw-port	0
alg-port	2400
audit-interval	20

nat-traversal

enabled

MGCP/NCS Configuration

Enter the following information to configure the **mgcp-config** element.

1. **private-realm**—Set the ID of the private realm (location of the media gateways). This private-realm field value must correspond to a valid identifier field entry in a realm-config. This is a required field.
 2. **private-address**—Set the IP address on the media interface in the private realm that the media gateways use as their call agent or softswitch IP address. This is a required field.
 3. **private-port**—Set the port number of the private realm’s network interface. The default value is **2727**. The valid range is:
 - Minimum—1025
 - Maximum—65535
 4. **public-ca-host**—Set the FQDN of the call agent located in the public realm. This field is optional.
 5. **public-ca-address**—Set the IPv4 address of the call agent or softswitch. This field is required.
 6. **public-ca-port**—Enter the public UDP Port of the call agent or softswitch. The default value is **2727**. The valid range is:
 - Minimum—1025
 - Maximum—65535
 7. **public-realm**—Enter the public realm of the call agent or softswitch. This is a required field. This public-realm field value corresponds to a valid identifier field entry in a realm-config that has already been configured.
 8. **public-gw-host**—Set the FQDN to use in the endpoint MGCP messages on the public side of the Net-Net SBC. If this field is left empty, the host part of the endpoint name will be the public gateway IP address (i.e., the public-gw-address field value). This field is optional.
 9. **public-gw-address**—Set the IPv4 address on the media interface in the public realm. This field value is the media gateway address that the Net-Net SBC uses to communicate with the call agent or softswitch. If this parameter is entered with a subnet mask in slash notation, 1:1 gateway mapping is enabled. This field is required. The default value is **0.0.0.0**.
 10. **public-gw-port**—Set the port on media interface in the public realm. This field value is the media gateway port that the Net-Net SBC uses to communicate with call agent or softswitch. The default value is **2427**. The valid range is:
 - Minimum—1025
 - Maximum—65535
 11. **second-public-gw-port**—Set the second UDP port on public-gw-address where the Net-Net SBC receives packets from the call agent or softswitch. The Net-Net SBC can receive messages from the call agent or softswitch on either the public-gw-port or the second-public-gw-port. The default value is **0**, meaning this feature is disabled. The valid range is:
 - Minimum—0, 1025
 - Maximum—65535

12. **audit-interval**—Set the interval in seconds between AUEP commands that the Net-Net SBC sends to the endpoint (gateway/IAD). The default value is **0**. The valid range is:
 - Minimum—0
 - Maximum—999999999
13. **nat-traversal**—Enable or disable whether or not the MGCP ALG assumes that all (gateway) endpoints are behind a NAT. The default value is **disabled**. Valid values are:
 - enabled | disabled
14. **alg-port**—Set the port used to send a packet from the network processor to the host processor. Each mgcp-config must have a unique ALG port so the ALG function can distinguish which mgcp-config element applies to packets sent up from the network processor. The default value is **2427**. The valid range is:
 - Minimum—1025
 - Maximum—65535

DNS Authentication

This section explains how to configure DNS authentication.

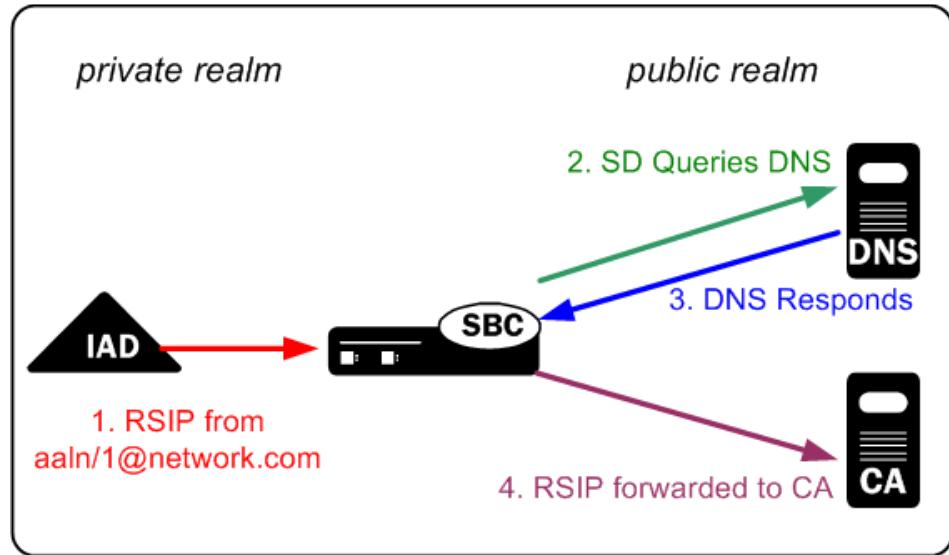
Some MGCP/NCS deployments require DNS authentication of endpoints for security purposes. The Net-Net SBC can perform DNS authentication against endpoint IPv4 addresses. This authentication checks an endpoint's domain name (layer 5) against the endpoint's IPv4 address received in a layer 3 message. If the two identifiers match, an original RSIP is forwarded to the call agent and the phone call can begin.

Note: The endpoint equipment and the DNS server must be compatible with the DNS authentication process.

In the diagram below, the following four steps take place:

1. The endpoint registers with an RSIP to the Net-Net SBC.
2. The Net-Net SBC queries a DNS server to check if the layer 5 endpoint name is the same as the layer 3 IPv4 address of the registering GW.
3. The DNS server responds to the Net-Net SBC indicating that the IPv4 address and domain name match.

4. The Net-Net SBC forwards the RSIP from the GW to the CA.



If the domain name and IPv4 address do not match, the Net-Net SBC issues an authentication failed 400 message back to the endpoint and the RSIP is not forwarded to the call agent.

DNS Authentication Configuration Overview

You need to configure the following elements for DNS authentication to work:

- Translation rules—unique sets of rules applied to incoming and outgoing calls. Translations are used to add, delete, or change character strings within an address.
- Session Translation—defines how translation rules are applied to both incoming and outgoing calls. Multiple translation rules can be referenced and applied using the Session Translation element, which groups rules together and allows them to be referenced by one name.

The Net-Net SBC applies the translation rules established in this field in the order in which they are entered. To enable DNS authentication without using any session translations, set up an empty session translation by leaving the in-translationid and out-translationid parameters blank. (See the Number Translation section to learn how to configure a translation rules and session translation elements.)

- DNS servers' IPv4 addresses must also be configured. These parameters are found in the network interface elements. You can configure as many as three DNS servers per network interface. Instructions on how to add DNS servers is located in the Network Interface section of this guide.

After the translation rules and sessions translation prerequisites have been set, set the parameters that enable DNS authentication for your MGCP/NCS configuration.

To configure DNS authentication for MGCP/NCS:

1. Set the **DNS Authentication** field to enabled.
2. Set the **DNS Translation** field to a configured session translation ID.

ACLI Instructions and Examples

This section describes how to configure the DNS authentication.

To configure DNS authentication for MGCP/NCS:

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)# **media-manager**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# **mgcp-config**
ACMEPACKET(mgcp-config)#[/b]

The following is an example what an MGCP/NCS DNS authentication configuration might look like. Parameters not described in this section are omitted below.

dns-authentication	enabled
dns-translation	Session-translation-ID

Set the following parameters to configure the **mgcp-config** element.

1. **dns-authentication**—Enable or disable the DNS authentication feature. The default value is **disabled**. Valid values are:
 - enabled | disabled
2. **dns-translation**—Enter a valid translation rules ID to use, i.e., what characters in the address will be added, replaced, or deleted. If you enable the MGCP DNS authentication feature, then this field is required. The value of this field must be a configured session translation.

Additional Parameters

In addition to entering parameters located in the MGCP configuration element, you need to configure the translation rules, session translation, and network interface elements. See the Number Translation and System Configuration sections for details.

MGCP/NCS Options

This section describes the MGCP/NCS features. The MGCP/NCS features are:

- Send Media Only
- X-Via Header Configuration

You configure these features using the MGCP/NCS options parameter.

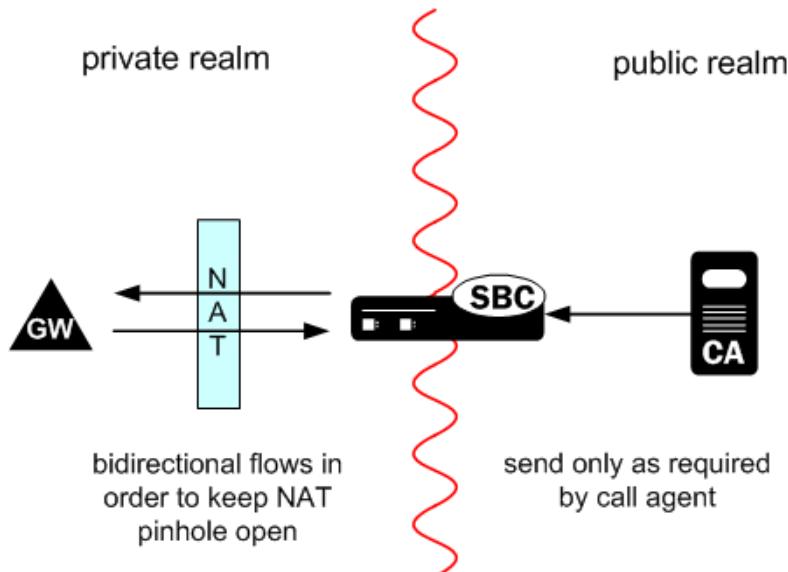
Send Media Only

In an MGCP/NCS deployment, network elements like announcement servers and media servers may be send-only devices in that they can never receive network traffic. If these send-only devices receive traffic, they might behave improperly causing unstable network conditions.

The MGCP/NCS send media only feature ensures that send-only devices never receive media traffic. When a gateway tries to contact a send-only MGCP/NCS element through a Net-Net SBC, the Net-Net SBC stops the gateway's traffic from reaching the send-only device. This commonly happens in a hosted NAT traversal situation. The Net-Net SBC needs the gateway to send data through the NAT in order to keep the pinhole open. However, if the data coming from the gateway is passed through the Net-Net SBC toward the send only device, undesirable consequences may result.

To configure MGCP/NCS send media only, set the options parameter as follows.

1. The send media only feature is configured in the options parameter.
 - **options drain-sendonly**



Enabling the send media only feature on the Net-Net SBC respects the send-only device's need to never receive traffic. At the same time, a gateway located behind a NAT is not treated as a send-only device. The gateway and Net-Net SBC communicate in a way that keeps the NAT pinhole open. All traffic received from the gateway is dropped at the Net-Net SBC and not forwarded toward the real Call Agent.

Signaling the Source IPv4 Address of Endpoints for 911 Services

The MGCP/NCS X-Via header enables the Net-Net SBC to handle 911 class calls. The requisite information is inserted in the MGCP/NCS message in the X-Via header in order that the CA can accept and act on 911 calls.

To configure MGCP/NCS X-Via header for 911 services, set the options parameter as follows.

1. The X-via header is configured in the options parameter. There are two ways to configure this.
 - **options x-via=endpoint**—This option signifies that the endpoint is either a router or a phone. The X-via header uses a GW's IPv4 address when GWs have public IPv4 addresses. The X-via header uses the NAT's public address when GWs use private addressing from behind a NAT.
 - **options x-via=both**—This option signifies that there are two addresses inserted into the X-via header, the private IPv4 address of the GW and the public IPv4 address of the Net-Net SBC.

Loose Authentication

When DNS authentication is enabled, you can force the Net-Net SBC to immediately start a call, before the caller is successfully authenticated. This mode works by assuming a successful authentication will be made and immediately sending a NTFY to a CA.

If the DNS query fails completely, for example the query can not reach the DNS server, then the call will still connect. If the DNS query returns an authentication failure, the AUEP (started above) to the endpoint will be stopped and all future NTFY message are not sent to the endpoint.

ACLI Instructions and Examples

This section describes how to configure the send media only and X-Via header configuration features in the MGCP configuration element.

To configure MGCP options:

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

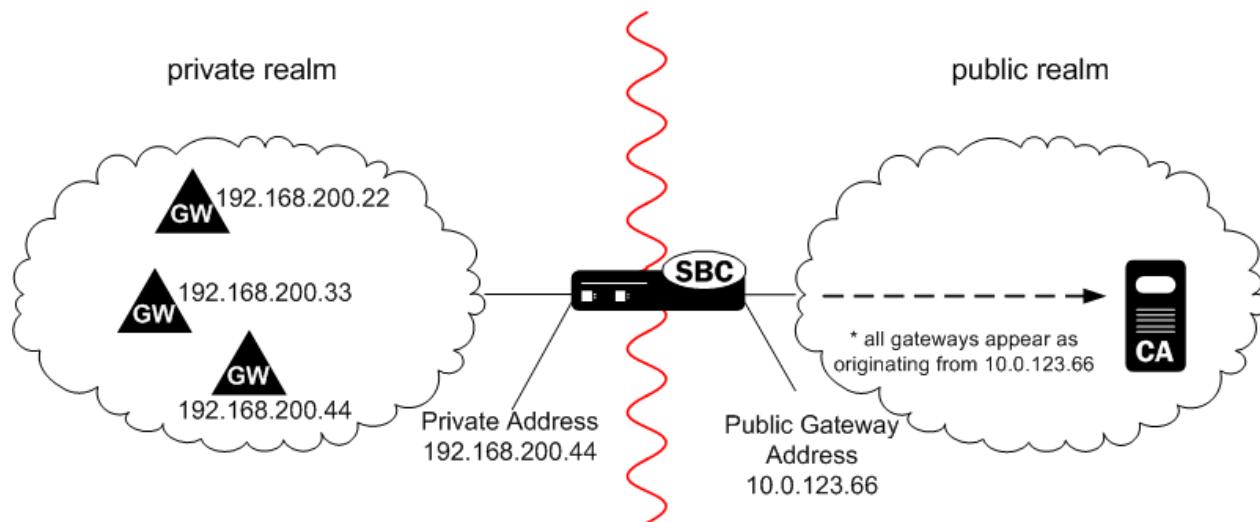
The following is an example what an MGCP option configuration might look like. Parameters not described in this section are omitted.

options	x-via=endpoint
----------------	-----------------------

MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints

In the typical MGCP/NCS deployment, several GWs that send their traffic through the Net-Net SBC exist in the private realm. These GWs are presented to the CA in the public realm as originating from the public gateway address on the Net-Net SBC. The public gateway address exists on the public-realm-facing network interface as a single IPv4 address and port pair.

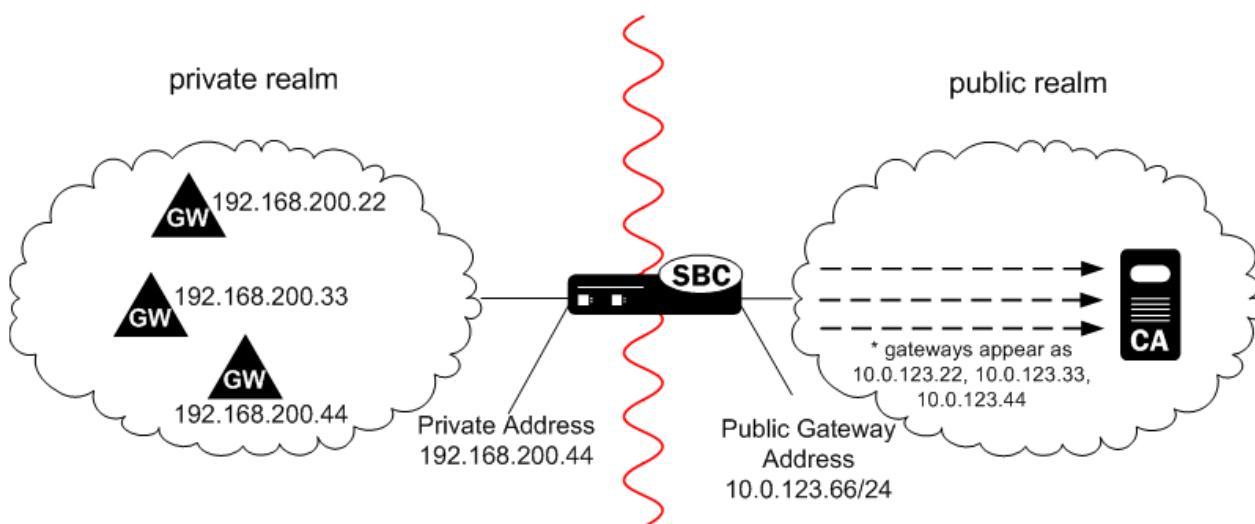
In such a configuration, the GWs believe that the private realm interface on the Net-Net SBC is the CA. Likewise, the CA believes that public realm interface on the Net-Net SBC is one or many GWs. The CA would see all traffic coming from the gateways as sourced from public gateway address. In the following diagram, the CA sees the traffic coming from the GWs as sourced from the public gateway address 10.0.123.66.



From the perspective of the CA, there is only one GW with which it communicates, even though there can be multiple GWs behind the Net-Net SBC.

In some MGCP/NCS deployments, the CA needs to interpret each GW as originating from a distinct IPv4 address. The Net-Net SBC can make each GW appear distinct by assigning it its own unique IPv4 address in the public realm.

The following diagram illustrates that when 1:1 IPv4 address mapping is enabled, all traffic originated in the private realm from the GWs appears with unique IPv4 addresses per gateway in the public realm.



Configuring the netmask on the Public Gateway Address parameter tells the Net-Net SBC how much of the GW's address to copy onto the public realm traffic. In the example, the final 8 bits of each GW is appended to the public gateway address's first 24 bits. This combination is used to present all GWs with unique IPv4 addresses to the CA.

MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints Configuration Overview

You configure the 1:1 IPv4 address mapping feature in the public gateway address parameter. When this parameter is configured this feature is disabled. When this parameter is configured with both an IPv4 address and netmask in slash notation, the feature is enabled.

When the public gateway address is set to an IPv4 address and netmask combination, all of the host bits (32 minus the netmask) are copied from the incoming source address in the packet to the outgoing source address in the packet. However, this only works if the source address comes from a network that has the same netmask or smaller as the public gateway address.

To enable MGCP/NCS 1:1 IPv4 address mapping for gateways and endpoints:

1. Configure an IPv4 address and netmask in slash notation for the public gateway address. If the netmask is omitted, the gateway masquerading function will be disabled. Make sure that this netmask has at least as many bits as the netmask for the IPv4 address on the interface of the private realm.

ACLI Instructions and Examples

This section describes how to configure MGCP/NCS 1:1 IPv4 address mapping for gateways and endpoints.

To configure MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# **medi a-manager**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(medi a-manager)# **mgcp-config**
ACMEPACKET(mgcp-config)#

The following is an example what an MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints configuration might look like. Parameters not described in this section are omitted below.

publ i c-gw-address	192. 168. 200. 0/24
---------------------	---------------------

MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints Configuration

Enter the following information to configure MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints.

1. **public-gw-address**—Set the IP address on the media interface in the public realm. This field value is the media gateway address that the Net-Net SBC uses to communicate with the call agents or softswitch. This is a required parameter. This parameter must be entered with a netmask in slash notation. The netmask should match that of the network you are masquerading in the private realm. The default value is **0.0.0.0**.

MGCP/NCS SuperNAT

The MGCP/NCS superNAT feature enables a Net-Net SBC configured to use the masquerading feature (MGCP/NCS 1:1 IPv4 Address Mapping for Gateways and Endpoints) to pass any message coming from the call agent to the gateway, even when the Net-Net SBC does not have an entry for that gateway.

If this feature is enabled and the masquerade feature is being used, the Net-Net SBC will allow the messages. If you do not enable this feature, then the Net-Net SBC only passes AUEP and DLCX messages from the call agent to the gateway if it does not have an entry for that gateway.

ACLI Instructions and Examples

To configure MGCP/NCS superNAT:

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(config)# **session-router**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **mgcp-config**
ACMEPACKET(mgcp-config)#
4. You can either add support to a new MGCP configuration or to an existing MGCP configuration:
 - 4a. If you do not currently have an MGCP configuration, you can add the option by typing options, a <Space> and then **super-nat**.
ACMEPACKET(mgcp-config)# **options super-nat**
 - 4b. Select the MGCP configuration so that you can add MGCP superNAT support to it. Then, to add this option to a list of options that you have already configured for the MGCP configuration, type **options** followed by a <Space>, the plus sign (+), and the **super-nat** option.
ACMEPACKET(mgcp-config)# **select**
ACMEPACKET(config)# **options +super-nat**
5. Save your work using the ACLI **save** or **done** command.

Endpoint Representation

An MGCP/NCS deployment can take advantage of various ways of representing endpoints to the call agent. Selecting the proper endpoint representation mode can solve the need for unique endpoint representation when endpoints are located behind NATs. Specialized endpoint representation can also be used for formatting endpoints for different call agent systems.

In a common VoIP scenario, multiple GWs with unique private IPv4 addresses exist behind a NAT. The identifiers of two devices might be aaln/1@192.168.200.11 and aaln/1@192.168.200.12 respectively. After packets from these two devices leave the NAT and enter public address space, bound for a Net-Net SBC, they both have the same endpoint ID. Thus, there is no way to distinguish traffic sent from or destined to each device. Traffic from both phones could look like aaln/1@63.168.127.12 if 63.168.127.12 was the public IPv4 address of the NAT. To distinguish traffic, the Net-Net SBC can create unique endpoint names.

Creating unique endpoint names involves reformatting the endpoint name of a GW. A unique identifier is calculated for each endpoint and then inserted into the full

endpoint name that remains intact after traversing a NAT. The unique identifier is calculated by using the mode and divisor fields, which are compliant with MGCP/NCS specifications.

Endpoint Number Computation

Endpoint not behind a NAT

The Net-Net SBC creates an endpoint identifier to insert into the layer-5 name of an endpoint when an appropriate mode option is chosen. Endpoint identifiers are computed differently if a GW is behind a NAT or not. In the following examples, the divisor is 65536.

The computation of an endpoint identifier for an IPv4 address not behind a NAT is as follows:

1. Multiply the decimal equivalent of the first octet of the IPv4 address by 256^3 , see column A in the following example.
2. Multiply the decimal equivalent of the second octet of the IPv4 address by 256^2 , see column B in the following example.
3. Multiply the decimal equivalent of the third octet of the IPv4 address by 256^1 , see column C in the following example.
4. Multiply the decimal equivalent of the fourth octet of the IPv4 address by 256^0 , see column D in the following example.
5. Add the numbers (A through D) computed in steps 1-4.
6. Calculate the modulus on the number computed in step 5 by the number chosen for the divisor (see next section) parameter. The result is the unique endpoint identifier.

GW not behind a NAT:

I AD IP Address:	192	168	45	12
	\times 256 ³	\times 256 ²	\times 256 ¹	\times 256 ⁰
	A	B	C	D

$(A+B+C+D) \% \text{ divisor} = \text{endpoint identifier}$

$3221225472 + 11010048 + 11520 + 12 = 3232247052$

$3232247052 \% 65536 = 11532$

Endpoint behind a NAT

The computation of an endpoint identifier for an IPv4 address behind a NAT is as follows:

1. Multiply the decimal equivalent of the third octet of the public IPv4 address by 256^3 , see column A in the following example.
2. Multiply the decimal equivalent of the fourth octet of the public IPv4 address by 256^2 , see column B in the following example.
3. Multiply the decimal equivalent of the third octet of the IPv4 address behind the NAT by 256^1 , see column C in the following example.
4. Multiply the decimal equivalent of the fourth octet of the IPv4 address behind the NAT by 256^0 , see column D in the following example.

5. Add the numbers (A through D) computed in steps 1-4.
6. Calculate the modulus on the number computed in step 5 above by the number chosen for the divisor (see next section) parameter. The result is the unique endpoint identifier.

GW behind a NAT

IAD IP Address:	192	168	45	12
			x 256 ¹	x 256 ⁰
Gateway IP Address:	10	11	12	13
			x 256 ³	x 256 ²

C D
A B

$$(A+B+C+D) \% \text{ divisor} = \text{endpoint identifier}$$

$$201326592 + 851968 + 11520 + 12 = 202190092$$

$$202190092 \% 65536 = 11532$$

Valid Divisors

Valid divisors are: 256^1 , 256^2 , 256^3 , 256^4 .

Endpoint Translation Mode

There are seven modes used to translate MGCP/NCS endpoint names. These modes specify the order and format of layer-5 endpoint names. Given an endpoint name entering a Net-Net SBC from the private MGCP realm, translation modes describe the format of the endpoint name as it exits the Net-Net SBC into the public realm.

For the following examples, the pre-SD endpoint name is described as:
`<endpoint>/<unit>@gateway`, where gateway is either an FQDN or IPv4 address.

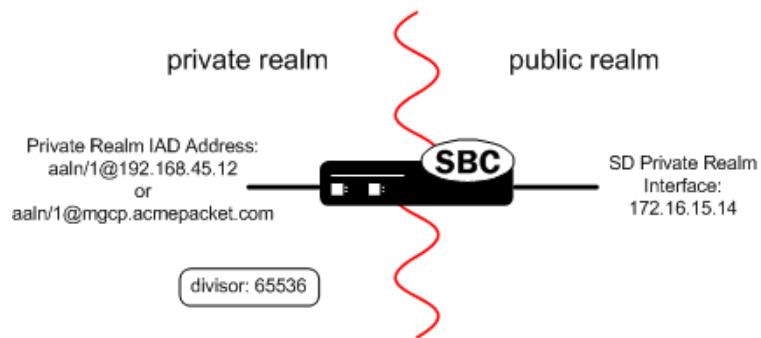
Mode	Description
None	The endpoint name will not be translated.
OnlyHost	Equivalent to using the None mode.
Host	<p>Inserts a term before the unit number in the endpoint name.</p> <p>If the gateway is an FQDN, the left-most part of the FQDN, after the left-most dot, is used as the unit term.</p> <p>If the gateway is an IPv4 address, the decimal equivalent of the IPv4 address is used as the unit term. (The decimal equivalent of an IPv4 address equals the A+B+C+D equation in the previous section).</p> <p>Example: <code><endpoint>/<unit-term><unit>@post.SD.IP</code></p>

Mode	Description
LinePrefix	<p>Used primarily in conjunction with a divisor field.</p> <p>This mode inserts the endpoint identifier immediately before the unit number.</p> <p>If the pre-Net-Net SBC endpoint is an FQDN, the most recently received Layer3 IPv4 address is used to compute the endpoint identifier.</p> <p>If the divisor field is left blank, the endpoint identifier will be the decimal equivalent of the IPv4 address.</p>
	Example: <endpoint>/<endpoint-identifier><unit>@post.SD.IP
LineUnit	<p>Used primarily in conjunction with a divisor field.</p> <p>This mode inserts the endpoint identifier immediately before the unit number. A slash is inserted between the endpoint identifier and the unit number.</p> <p>If the pre-Net-Net SBC endpoint is an FQDN, the most recently received Layer3 IPv4 address will be used to compute the endpoint identifier.</p> <p>If the divisor field is left blank, the endpoint identifier is the decimal equivalent of the IPv4 address.</p>
	Example: <endpoint>/<endpoint-identifier>/<unit>@post.SD.IP
FQDN	<p>The FQDN is surrounded by slashes and inserted between the endpoint and unit number. In addition, the dots are removed from the FQDN.</p> <p>If the endpoint is identified by IPv4 address, its decimal equivalent is inserted in the FQDN position.</p>
	Example: <endpoint>/<FQDNwithoutdots>/<unit>@post.SD.IP
FQDN2	<p>The FQDN is inserted between the endpoint and unit number with slashes on either side of it. Note that the difference between FQDN2 and FQDN modes is that the dots are NOT removed from the FQDN in FQDN2 mode.</p> <p>If the endpoint is identified by IPv4 address, its decimal equivalent is inserted in the FQDN position.</p>
	Example: <endpoint>/<FQDN>/<unit>@post.SD.IP

Endpoint Translation Examples

In the following figure, all modes and address representations are described according to the mode chosen.

Example for MGCP Mode Options



Endpoint Translation	In the following examples, the divisor is 65536. Therefore, the endpoint identifier is 11532.
-----------------------------	---

Mode	FQDN Addressing	IP Addressing	
None	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Post-SD
Host	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/mgcp1@172.16.15.14	aaln/115321@192.168.45.12 aaln/7557726841@192.168.45.12 *	Post-SD
LinePrefix	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/115321@172.16.15.14 aaln/7557726841@192.168.45.12 *	aaln/115321@172.16.15.14 aaln/7557726841@192.168.45.12 *	Post-SD
LineUnit	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/11532/1@172.16.15.14 aaln/7557726842/1@172.16.15.14 *	aaln/11532/1@172.16.15.14 aaln/7557726842/1@172.16.15.14 *	Post-SD
FQDN	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/mgcp.acmepacket.com/1@172.16.15.14	aaln/755772684/1@192.168.45.12	Post-SD
FQDN2	aaln/1@mgcp.acmepacket.com	aaln/1@192.168.45.12	Pre-SD
	aaln/mgcp.acmepacket.com/1@172.16.15.14	aaln/755772684/1@192.168.45.12	Post-SD

* denotes that the divisor parameter was not configured.

Unit Prefix

You can configure a prefix string on the unit term in an MGCP address. This prefix is configured with the unit-prefix parameter in the MGCP configuration element. A unit-prefix can be used as a tag to mark all calls originating in a specific realm. Unit prefixes are used primarily for routing purposes.

To configure the endpoint translation:

1. Set the mode parameter according to your needs.
2. Set the divisor or unit prefix parameters if applicable.

ACLI Instructions and Examples

To configure endpoint translation information:

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(config)# **session-router**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **mgcp-config**
ACMEPACKET(mgcp-config)#[/list]

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

```
mode          onI yhost
di vi sor   65536
uni t-prefix x
```

Endpoint Translation Configuration

Set the following parameters to configure the endpoint translation:

1. **mode**—Set the MGCP-NAT mode. This field defines how endpoint names are translated as MGCP flows traverse the Net-Net SBC. This is a required field. The default value is **Lineunit**. Valid values are:
 - **Host**—A “unit” term is added to endpoint name on public side to uniquely identify the gateway/host on the private side. The left-most part of the private FQDN is used as the unit term (or unit name).
 - **LinePrefix**—Divisor field value is used to compute a number to insert into the localname part of the endpoint name. The number to be inserted is the IP address modulo the divisor. This mode inserts this number before the channel number. Example: aaln/1 becomes aaln/1231. The IP address part is replaced by the public-gw-address.
 - **LineUnit**—Divisor field value is used to compute a number to insert into localname part of endpoint name. The number inserted is the IP address modulo the divisor. This mode adds the unit-number term defined in the conventions section of <ftp://ftp.rfc-editor.org/in-notes/rfc3435.txt> (e.g., aaln/2 becomes aaln/123/2). The IP address part is replaced by the public-gw-address (also defined in this element).
 - **FQDN**—Dots are removed from the host portion of the private endpoint. Example: the address aaln/2@abc.xyz.com on the private (i.e., gateway) side would become aaln/abcxyzcom/2@sd.com on the public (i.e., call agent) side.
 - **FQDN2**—Dots are retained in the host portion of the private endpoint. Example: the address aaln/2@abc.xyz.com on the private (i.e., gateway) side would become aaln/abc.xyz.com/2@sd.com on the public (i.e., call agent) side.
 - **OnlyHost**—Endpoint name is not translated.
 - **None**—Endpoint name is not translated.
2. **divisor**—Set the divisor for use with calculating an endpoint identifier. This field is used to determine the number for the LinePrefix or LineUnit. The remainder of the private IP address divided by this number becomes the prefix/unit number. If FQDNs are used for network addressing, this field is not used. The default value is **256**. Valid values are:
 - 256 | 65535 | 16777216 | 4294967296
3. **unit-prefix**—Enter the prefix for the unit term of the endpoint name. For modes that add a unit term to the user part of the endpoint name, this field value is placed in front of the unit number or name when creating a public endpoint name.

Call Agent Redundancy

MGCP/NCS CA redundancy can be provisioned on a Net-Net SBC to enhance its HA properties. In the event of a call agent failure, all transactions are directed to the next provisioned call agent as configured.

Call Agent Redundancy Configuration Overview

Call agent redundancy enhances the high availability of the Net-Net SBC MGCP proxy by allowing it to communicate to a secondary (or tertiary) call agent in the event that the primary call agent becomes unavailable. In the event of a CA switch over, all current and future transactions will be directed to the next provisioned CA.

CA redundancy works by incorporating DNS capability into the current MGCP/NCS implementation, and developing a ping mechanism capable of monitoring the health of a CA.

To configure CA failover, you must specify the provisioned MGCP/NCS CA as a FQDN instead of an IPv4 address. If the resolved FQDN returns more than one IPv4 address (valid "A" record), then CA redundancy will be considered activated.

Furthermore, you can configure the Net-Net SBC to ping the CA a specified number of times before declaring the CA out of service. This function is useful in networks where congestion becomes an issue and a CA may not respond initially due to a delayed response time for an RSIP.

Defining a Set of CAs for Redundancy

You can also manually configure individual IP addresses for the Net-Net SBC to use for provisioning redundant MGCP Call Agents (CA). The CA failover IP address parameter defines an MGCP configuration's set of redundant CAs, explicitly by IP addresses that you specify. When you do not manually configure the IP addresses, the Net-Net SBC learns of multiple CAs as returned in a DNS query.

If you configure a hostname for the public CA host and the CA redundancy parameter is enabled, the Net-Net SBC will use the IP addresses returned in the DNS response to populate its list of multiple CAs. Even though you might also configure the CA failover IP list, it will be ignored because the redundancy mechanism initiated by configuring a CA hostname takes precedence.

The order in which you configure the CA Failover IP list indicates the order in which the Net-Net SBC attempts to use each CA.

DNS Resolution for Call Agent Redundancy

If the CA redundancy feature is enabled and you do not manually define a set of CAs, the call agent is configured using its FQDN instead of its IP address. If both addressing methods are configured and DNS resolution on the FQDN is successful, address(es) returned by the DNS will take precedence.

At boot time or after an **activate config** command is issued, MGCP queries the DNS associated with the public realm network interface where the MGCP proxy exists. DNS should return at least one (and more than one if CA redundancy is desired) "A" record that identifies the IPv4 address of the primary, secondary, tertiary, and so on. The first IPv4 address returned by the DNS becomes the initial Call Agent and the others, in the order returned by the DNS, are used sequentially in failover situations.

The group of CA IP addresses returned by the DNS is known as the call agent group. The call agent group is associated with a user-defined refresh timer. The Net-Net SBC pings the members of the call agent group according to the CA Ping Interval to check that the call agents are still reachable and active.

Call Agent Failover

Call agents can fail over 3 ways. The two scenarios where the active call agent initiates the switch are:

1. Explicit—The currently active CA notifies the SD to use a different CA, also known as the notified entity. The notification method only works if the CA specified is in the form of an IPv4 address.
2. Implicit—The currently active CA address no longer matches the source address in an incoming packet from the CA. In this case the SD will fail over and the source address will become the new CA.

The scenario where the Net-Net SBC initiates the request is:

3. PING timeout—When a CA fails to respond within the period defined in the call agent ping interval, it is considered to be out-of-service. If additional in-service call agents exist in the call agent group, the next CA will become the active call agent.

After a CA switchover, the Net-Net SBC will not initiate another switchover to return to the original state. It will only switch based on a failure. The only two situations that will force a switchover a second time are if the CA explicitly requests a switchover or if the newly active CA goes out of service.

To add CA redundancy to an MGCP/NCS configuration, populate the following parameters:

1. **public-ca-host**—This parameter is only required when CA redundancy is configured. Available values are:
 - Hostname of the CA (that is identified by the required public-ca-address).
 - Hostname identifying a CA redundancy group. In order to identify an address as a CA redundancy group, it must be entered in FQDN format.
2. **ca-ping-interval**—This parameter is required when CA redundancy is configured. The CA ping interval sets the amount of time in seconds that the Net-Net SBC waits to send a ping to the active call agent to determine if it is still healthy. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—999999999
3. **ca-redundancy**—This parameter indicates if CA redundancy is enabled. If this field is set to **disabled**, the Net-Net SBC operates with a single CA as configured in the Public Call Agent Host parameter. The default value is **disabled**. Valid values are:
 - enabled | disabled
4. **ca-ping-method**—Enter the ping method used for call agent redundancy. This parameter is the prototype of a ping method sent to a call agent to determine its state. This parameter specifies any legal GW-originated message. After the Net-Net SBC replaces the sequence number in the given prototype message, the message is sent to the current CA. When a response is not received from the CA, it is assumed to be out-of-service. A valid prototype message could be:
`NTFY [100] aal n/1@172. 16. 2. 1`

5. **ca-ping-retries**—Enter the number of times you want the Net-Net SBC to try to ping a call agent before it determines the call agent is out of service.

The default value is 0. The valid range is:

- Minimum—0
- Maximum—999999999

ACLI Instructions and Examples

To configure call agent redundancy 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 system-level configuration elements.
ACMEPACKET(config)# **session-router**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

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

The following is an example what an MGCP/NCS CA Redundancy configuration might look like. Parameters not described in this section are omitted below.

```
ACMEPACKET(mgcp-config)# show
      ca-redundancy          enabled
      ca-ping-method         NTFY [100] aal n/1@172.16.2.1
      ca-ping-interval       4
ACMEPACKET(mgcp-config)#{
```

CA Redundancy Configuration

Set the following parameters to configure call agent redundancy:

1. **ca-redundancy**—Enable or disable the call agent redundancy feature. The default value is **disabled**. Valid values are:
 - enabled | disabled
2. **ca-ping-method**—Set the CA ping method to a valid prototype such as **NTFY [100] aal n/1@172.16.2.1**
3. **ca-ping-interval**—Set the amount of time in seconds between pings sent to the CA to check for health. The default value for this parameter is **0**. The valid range is:
 - Minimum—**0**
 - Maximum—**999999999**

Manually Defining a Set of CAs for Redundancy

To specify a call agent address(es) to support MGCP call agent redundancy:

1. **ca-failover-ip-addresses**—Enter a list of IP addresses for call agent redundancy support. You must enter the list of IP addresses enclosed in parentheses and separate each IP address with a <Space>.

```
ACMEPACKET(mgcp-config)# ca-failover-ip-addresses (192.168.24.2
192.168.24.3 192.168.24.4)
```

Enhanced Roaming (IP Address Carrying)

The IP Address Carrying feature allows multiple active registrations for the same user at different locations. The locations are distinguished by their private IP address or, if necessary, the public address of the firewall the device is located behind.

This feature is implemented by including the IP addresses and port of the endpoint or firewall in two new fields included in a MGCP (or SIP) message sent from the Net-Net SBC.

For an MGCP application, the Net-Net SBC adds the following two headers in MGCP messages:

```
X-Usradd=192.168.1.10  
X-Uspprt=2427
```

where

- X-Useradd is the MGCP equivalent of the SIP useradd field.
- X-Uspprt is the MGCP equivalent of the SIP userport field.

ACLI Instructions and Examples

To configure MGCP IP Address Carrying:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the media-specific configuration elements.
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# mgcp-config
ACMEPACKET(mgcp-config)#
4. Type **options x-user-info** and press <Enter>. This completes the configuration of MGCP IP Address Carrying.
ACMEPACKET(mgcp-config)# options x-user-info.

MGCP Sans Media

The MGCP sans media feature lets MGCP calls without media being managed by the Net-Net SBC to work.

In prior releases, the Net-Net SBC always managed the media for MGCP, you could not configure it to do otherwise. With this release, you can set the media manager's state parameter to disabled to let the Net-Net SBC pass SDP to the endpoints without managing the media.

Configuring MGCP Sans Media	You can configure MGCP sans media using the ACLI or Net-Net EMS.
------------------------------------	--

ACLI Instructions and Examples

To configure MGCP sans 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-level configuration elements.
ACMEPACKET(configure)# **media-manager**
ACMEPACKET(media-manager)#
 3. Type **media-manager** and press <Enter> to access the media-manager parameters.
ACMEPACKET(media-manager)# **media-manager**
ACMEPACKET(media-manager-config)#
 4. **state**—Disable this feature to let the Net-Net SBC pass SDP to the endpoints without managing the media. The default value is **enabled**. The valid values are:
 - enabled | disabled

MGCP Congestion Control

MGCP congestion control is designed to help customers handle large call events in an oversubscribed environment. When you enable this feature, the Net-Net SBC can send a system busy message back to the call agent for new calls when system resources have been exhausted.

Overload conditions are determined by CPU utilization. Using a new option in the media manager configuration, you can set the threshold that defines the point at which overload occurs. When the threshold is exceeded, the Net-Net SBC issues an alarm and a corresponding trap.

How It Works

In the media manager configuration, you can set a new option called **al gd-load-limit** that permits you to set the amount of CPU utilization for the threshold. The unit of the value you specify is a percentage of the CPU utilization; it cannot be set to a value less than zero or greater than one hundred. However:

- If you set this option to an invalid value below one hundred, then the Net-Net SBC uses a default of 95.
- If you set this option to an invalid value over one hundred, then the Net-Net SBC uses 100 as the value.

The values that you set apply to the total CPU utilization for all Net-Net SBC application tasks that run at a priority of 80 or more.

If the Net-Net SBC's CPU utilization equals or exceeds the threshold you configure, the Net-Net SBC will reject calls (off-hook NTFY messages) by sending 403 messages. The "off-hook message" is the only message that the Net-Net SBC rejects with a 403 message. And the Net-Net SBC re-sends 403 Intermediary Failure messages for subsequent retransmissions of calls that the Net-Net SBC has already rejected. CRCX and RSIP messages are not rejected, but instead are handled the same way they were prior to the implementation of MGCP congestion control. In addition, the Net-Net SBC tracks the number of NTFY Overload 403 Sent messages, which you can view using the ACLI **show algd NTFY** command.

When the CPU utilization falls below the threshold, the Net-Net SBC resumes accepting calls after the 60-second guard timer elapses and the trap clears.

Alarm Information

The Net-Net SBC sends notification using an SNMP trap and critical alarm that the CPU threshold has been exceeded. The trap and alarm are cleared when the CPU utilization returns to acceptable limits for a minimum of 60 seconds. The 60-second guard timer prevents the Net-Net SBC from oscillating in and out of the overload condition.

ACLI Instructions and Examples

The minimum value for the new **al gd-load-lim t** is 0, and the maximum is 100. Refer to the [How It Works \(715\)](#) section above for descriptions of system behavior when invalid values are configured.

To configure MGCP congestion control:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config terminal**
2. Type **media-manager** and press <Enter> to access the **media-manager** path.
ACMEPACKET(config)# **media-manager**
3. Type **media-manager** again 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)#
If you are adding support for this feature to a pre-existing media manager configuration, then use the ACLI **select** command to start editing the configuration.
4. **options**—Set the options parameter by typing **options**, a <Space>, the plus sign, and the option **al gd-load-lim t=X** (where X is the threshold for CPU utilization). Then press <Enter>.
ACMEPACKET(media-manager-config)# **options +al gd-load-lim t=90**
If you type **options al gd-load-lim t=X**, you will overwrite any previously configured options. In order to append the new option to the **media-manager**'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.

MGCP Restricted Latching

The Net-Net SBC supports restricted media latching for MGCP. Restricted latching offers security from rogue RTP packets by isolating the valid sources of RTP flows to well-known, signaled addresses.

When restricted media latching is enabled, the Net-Net SBC latches only to media from known source IP addresses in order to learn and latch onto the dynamic UDP port number. The IP address's origin can be either the SDP information or the SIP message's Layer 3 IP address, depending on your configuration.

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 the address range based on the received L3 IP address of the offer or answer.

For more information about restricted latching (and its relationship to symmetric latching), refer to the *Realms and Nested Realms* chapter of the *Net-Net Configuration Guide*. This feature does not have any impact on the **latching** parameter in the media manager configuration.

ACLI Instructions and Examples

This section explains how to configure restricted latching on the Net-Net SBC. These are the same parameters previously used for SIP only, but now they also support this feature for MGCP.

To configure restricted latching:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config ure terminal**
2. Type **media-manager** and press <Enter>.
ACMEPACKET# **medi a-manager**
ACMEPACKET(medi a-manager)#
 3. Type **realm-config** and press <Enter>.
ACMEPACKET(medi a-manager)# **real m-confi g**
ACMEPACKET(real m-confi g)#
 4. If you are adding support to a pre-existing realm, select the realm where you want to apply this feature.
ACMEPACKET(real m-confi g)# **sel ect**
identi fier:
1: Acme_Realm <none>
2: MGCP_Realm <none>
3: H323_Realm <none>
 5. **restricted-latching**—Enter the restricted latching mode. The default value is **none**. Valid values are:
 - **none**—Use no latching
 - **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. You only need to set this parameter if the restricted latching parameter is set to either **sdp** or **peer-ip**. The default is 32. The valid range is:
 - Minimum—1
 - Maximum—32
 7. Save and activate your configuration.

MGCP Endpoint Aging

When an MGCP gateway registers through the Net-Net SBC, memory is allocated to hold state information associated with each active endpoint on the gateway. In a number of circumstances this memory is never released. An endpoint aging mechanism has been added to the Session Director to remove state information that is no longer needed and free up associated memory.

How It Works

The Net-Net SBC maintains a per-endpoint timer to track when traffic was last received from the gateway. If the timer expires, the Net-Net SBC deletes the endpoint and frees its resources. If all endpoints associated with a gateway are deleted, then the Net-Net SBC removes the gateway entry, too.

Any traffic received from an endpoint resets the activity timer for that endpoint.

Dynamic Reconfiguration

This feature is RTC-supported. The Net-Net SBC applies newly configured values when an inactivity timer expires or when it creates a new endpoint entry.

However, when an activity timer is currently set to 0 or being set to 0, the Net-Net SBC sets all inactivity timers to the new value. However, the Net-Net SBC staggers the inactivity timers because it is undesirable to have a large number of endpoints deleted at the same time.

Considerations for HA

Inactivity timers are not replicated across the Net-Net SBCs in an HA node. If there is a switchover, the inactivity timers for all endpoints are initialized for the amount of time you set. However, the Net-Net SBC staggers the inactivity timers because it is undesirable to have a large number of endpoints deleted at the same time.

Deletion Smoothing

Built into this feature is a smoothing mechanism that, when necessary, staggers the deletion of endpoints to provide continuity in system resource use.

When you enable this feature, you set two parameters. One is the time value for the number of seconds after which an endpoint is considered inactive and is deleted. A second time value defines how many milliseconds to add to an endpoint's inactivity timer in order to stagger its time from the previous timer.

Note the following:

- If Endpoint 1 and Endpoint 2 send messages outside the window defined in the inactivity time rate, then the Net-Net SBC does not stagger deletions.
- If the MGCP endpoint sends a message (any kind of MGCP message) within the allowable time, then the Net-Net SBC resets the timer for that endpoint.

ACLI Instructions and Examples

You enable MGCP endpoint aging on the Net-Net SBC by setting two new options in the MGCP configuration:

- **ep-inactivity-timer=X**—Sets the amount of time in seconds before an MGCP endpoint expires; to stagger deletions, the **ep-inactivity-timer-rate** value can be added to the inactivity timer if multiple endpoints send messages within the inactivity timer rate window
- **ep-inactivity-timer-rate=X**—Defines a minimum gap in milliseconds between timers so that multiple expiring timers do not cause a CPU utilization spike; if you do not set a time, the Net-Net SBC uses a 100 millisecond default

To enable MGCP endpoint aging:

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 **mgcp-config** and press <Enter>.

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

```

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

4. **options**—Set the options parameter by typing **options**, a <Space>, and then the option name **ep-inactivity-timer=X** with a “plus” sign in front of it (where X is the time in seconds for the timer). Then press <Enter>.

At the system prompt, again type **options**, and then the option name **ep-inactivity-timer-rate=X** with a “plus” sign in front of it (where X is the window time in milliseconds and the time used to stagger deletions). Then press <Enter>.

```
ACMEPACKET(mgcp-config)# options +ep-inactivity-timer=100
ACMEPACKET(mgcp-config)# options +ep-inactivity-timer-rate=200
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

MGCP Stateful Graceful Backoff

The MGCP congestion control feature helps customers handle large call events in an oversubscribed environment. When enabled, the Net-Net SBC sends a system busy message back to the call agent for new calls when system resources have been exhausted. Overload conditions are determined by CPU utilization. When the threshold is exceeded, the Net-Net SBC issues an alarm and a corresponding trap.

Because some users have experienced a “ghost ring” (indefinite ringing on the side of the caller and dead air on the side of the called person when he answers), a new feature called MGCP graceful stateful backoff has been developed. In the previous MGCP congestion control implementation, the Net-Net SBC did not distinguish between NTFY messages from the calling party and the called party.

How It Works

When this feature is enabled, the Net-Net SBC creates a list of calls rejected due to overload. When a call is rejected, the Net-Net SBC checks both the request identifier and endpoint IP addresses to see if that particular call was rejected due to overload. If a match exists and a call has been rejected for this reason, the call is passed and no ghost ring occurs.

To prevent against memory exhaustion, the records the Net-Net SBC stores in the list it uses to match against have a limited lifetime. The default time for the life of a record on this list is two minutes, but you can set an option in the media manager configuration to the amount of time you require between 10 seconds and 10 minutes.

ACLI Instructions and Examples

Two new options have been added to the media manager configuration to support stateful graceful backoff:

- **stateful -mgbo**—Enables this feature, and specifies the stateful graceful backoff method:
 - **rejecthu**—The Net-Net SBC rejects NTFY(hu) messages that contain the request identifier of a previous NTFY(hd) it rejected
 - **accepthd**—The Net-Net SBC does not reject NTFY(hd) messages that contain a RQNT with ringing request identifier
- **mgbo-ti meout**—Sets the amount of time in milliseconds that the Net-Net SBC keeps records of calls rejected because the CPU utilization threshold was exceeded; valid range is 10 seconds to 10 minutes, with two minutes as the default

The **al gd-load-lim t** option must be configured in order to configure this MGCP congestion control enhancement. Refer to the MGCP Congestion Control section above for configuration instructions and examples.

To configure MGCP graceful stateful backoff:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **media-manager** and press <Enter> to access the **media-manager** path.
ACMEPACKET(config)# media-manager
3. Type **media-manager** again 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)#
```

If you are adding support for this feature to a pre-existing media manager configuration, then use the ACLI **select** command to start editing the configuration.

4. **options**—Set the options parameter by typing **options**, a <Space>, the plus sign, the option **stateful -mgbo** followed by a colon (:) and the stateful graceful backoff method you want to use (**rejecthu** or **accepthd**), and then press <Enter>. If you want to set a timeout for the record storage, type **options**, a <Space>, the plus sign, the option **mgbo-ti meout** followed by the equal sign (=) and the amount of time in milliseconds, and then press <Enter>.

```
ACMEPACKET(media-manager-config)# options +stateful -mgbo: accepthd
ACMEPACKET(media-manager-config)# options +mgbo-ti meout: 120000
```

If you enter these options without using 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.

MGCP Configurable CPU Sample Rate

An enhancement to the Net-Net SBC's [MGCP Congestion Control \(715\)](#) functionality, the configurable sampling rate prevents CPU resource exhaustion by allowing you to set the rate at which you want CPU measured.

To perform MGCP congestion control, the Net-Net SBC periodically measures CPU utilization. Without setting the CPU sampling rate, the preset for this measurement is ten seconds during normal usage states and five seconds during overload conditions.

Several options in the media manager configuration allow you to set the measurement interval to a time between one and ten seconds. For overload conditions, this measurement interval is set to five seconds if the value you set is greater than five. There are also new options that allow you to set parameters around rejection percentages and standby time.

How It Works

The media manager configuration options supporting this feature allow you to set the measurement interval for CPU sampling, rejections percentages, and the state of the Net-Net SBC. Some of these value are used in Net-Net SBC calculations that guard against sudden drops in rejection rate.

- `al gd-load-measurement-interval`—This value controls how often the Net-Net SBC's MGCP task checks the CPU utilization of application tasks. A small interval makes the task respond more quickly to changes in CPU utilization, but adds overhead to the CPU utilization of the task individually and to the overall CPU utilization of application tasks.

The valid range for this option is one to ten seconds. During an overload condition, this rate automatically defaults to five seconds if the value you set is greater than five seconds. All subsequent options described in this section depend on this option.

- `al gd-load-average-number-periods`—When you define this option, the Net-Net SBC's MGCP task also incorporates the average CPU utilization into its decision to enter congestion state (and start sending 403 messages to new NTFY off-hook commands). The Net-Net SBC computes average CPU utilization over the period of time equal to `al gd-load-measurement-interval` multiplied by the `al gd-load-average-number-periods` value. The MGCP task enters congestion state when both the current CPU utilization and the average CPU utilization exceed the `al gd-load-limit` value.
- `al gd-load-reduction-pct`—When you define this option, the Net-Net SBC uses its value to compute a new rejection rate if there is a decrease in CPU utilization during congestion state and standby state. The new rejection rate equals 100 minus the `al gd-load-reduction-pct` value, as a percentage of the previous rejection rate.

For example, if the previous rejection rate is 10% and the `al gd-load-reduction-pct` value is 10%, then the new rejection rate is 90% of 10%, which is 9%.

The valid range is 1 to 100, with a default of 0.

- **al gd-l oad-mi ni mum-rej ecti on-rate**—When you define the **al gd-l oad-reducti on-pct**, this options specifies the minimum rejection rate when computed using the **al gd-l oad-reducti on-pct** value. If a new rejection rate is less than the value set using this option, then the Net-Net SBC sets the new rejection rate to 0. The default is 5.
- **al gd-l oad-standby-peri od**—Time in seconds that the Net-Net SBC stays in a standby state after CPU utilization has receded below the load limit and before it goes back to the normal state. If overload conditions return, the Net-Net SBC returns to the overload state.

The valid range is 1 to 60, with a default of 60.

ACLI Instructions and Examples

To set the appropriate options for configurable CPU measurement intervals:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config ure terminal
 2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# medi a-manager
 3. Type **media-manager** again and press <Enter>.
ACMEPACKET(medi a-manager)# medi a-manager
ACMEPACKET(medi a-manager-confi g)#
 4. **options**—Set the options parameter by typing **options**, a <Space>, the options in the sample below (with a time or percentage value of your choosing), each with a “plus” sign in front of it, and then press <Enter>.
ACMEPACKET(medi a-manager-confi g)# options +al gd-l oad-measurement-interval=6
ACMEPACKET(medi a-manager-confi g)# options +al gd-l oad-average-number-periods=2
ACMEPACKET(medi a-manager-confi g)# options +al gd-l oad-reducti on-pct=50
ACMEPACKET(medi a-manager-confi g)# options +al gd-l oad-mi ni mum-rejection-rate=25
ACMEPACKET(medi a-manager-confi g)# options +al gd-l oad-standby-period=30
- 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.

MGCP/NCS X-Keepalives

Some VoIP phones have a *keep-alive* option to support intermediate NATs. This option involves the periodic transmission of RSIPs that include an X-Keepalive parameter.

The Net-Net SBC's default method of performing Hosted NAT Traversal (HNT) in MGCP/NCS environments is to send periodic AUEP messages to each endpoint. If it is sent prior to the NAT binding time-to-live (TTL) expiry time, this traffic refreshes the binding on the intermediary NAT device. This HNT technique is referred to as a "push"; that is, the hosted device pushes keepalive messages to each endpoint.

When the Net-Net SBC receives an RSIP with an X-Keepalive header present, it checks to see if that endpoint is known.

- If it is known, the Net-Net SBC responds with a 200 OK directly and suppresses the AUEP keepalives that are generated.
- If it is not known, the Net-Net SBC replaces the X-Keepalive header with Restart and forwards it to the call agent.

MGCP AUEP Suppression

In certain networks, gateways send RSIP messages with X-Keepalive restart method (RM) parameters as a means of keeping a NAT refreshed. The Net-Net SBC does not forward these messages, and in such cases it is redundant for the Net-Net SBC to send audit endpoint (AUEP) messages, even if it is enabled to do so.

The Net-Net SBC suppresses audit endpoint (AUEP) messages for gateways that send X-Keepalives. When the Net-Net SBC receives an X-Keepalive RSIP from any endpoint on a gateway, it will not send an AUEP message until it receives an RSIP with the RM parameter.

ACLI Instructions and Examples

To configure MGCP/NCS x-keepalives:

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 **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **mgcp-config**
ACMEPACKET(mgcp-config)#
4. You can either add support to a new MGCP configuration or to an existing MGCP configuration:
 - 4a. If you do not currently have an MGCP configuration, you can add the option by typing **options**, a <Space>, and then **x-keep-alive**.
ACMEPACKET(mgcp-config)# **options x-keep-alive**
 - 4b. Select the MGCP configuration so that you can add MGCP x-keepalive support to it. Then, to add this option to a list of options that you have already configured for the MGCP configuration, type **options** followed by a <Space>, the plus sign (+), and the **x-keepalive** option.
ACMEPACKET(mgcp-config)# **select**
ACMEPACKET(mgcp-config)# **options +x-keepalive**

5. Save your work using the ACLI **save** or **done** command.

MGCP Endpoint Aging Optimization

When you use MGCP endpoint aging in releases prior to Net-Net OS C5.1, the endpoint clean-up can consume a great deal of Net-Net SBC resources and so slow call processing. In Net-Net Release C5.1, this issue has been addressed by making the endpoint timers controlled (rather than actual). This way, the Net-Net SBC controls the rate at which endpoints age and how many endpoints to age, and keeps CPU load to a minimum. If the CPU load exceeds the threshold, controlled timers automatically reduce the rate of MGCP endpoint aging.

How It Works

Certain options in the media manager configuration use default values and are configurable so that the Net-Net SBC can perform MGCP endpoint aging optimization. For the options to apply, you must have enabled CPU limiting and inactivity timers.

These options determine how many endpoints are checked during the two-second controlled timer period, and also how many endpoints the Net-Net SBC is allowed to age during that loop. They also set the CPU threshold and define how many endpoints are checked and how many can be aged when the threshold is exceeded.

The options you can configure are defined in the following table.

Option Name	Default Value	Description
alg-aging-max-loop	100 (integer)	Maximum number of endpoints examined for every controlled timer tick (two seconds) to determine aging
alg-aging-max-exp	15 (integer)	Maximum number of endpoints the Net-Net SBC will age per loop, limits a flood of endpoints from being aged at the same time
alg-aging-cpu-limit	30 (integer)	Threshold for percentage of CPU; if CPU utilization exceeds this threshold, alg-aging-load-lax-loop and alg-aging-load-max-exp are used
alg-aging-load-max-loop	50 (integer)	When CPU threshold has been exceeded, maximum number of endpoints examined for every controlled timer tick (two seconds) to determine aging
alg-aging-load-max-exp	3 (integer)	When CPU threshold has been exceeded, acceptable maximum number of endpoints the Net-Net SBC will age per loop, limits a flood of endpoints from being aged at the same time

ACLI Instructions and Examples

Note that Acme Packet recommends that you use the default value in place for this feature. If you change these values from their defaults, do so with caution because faulty values can degrade system performance.

To set the options controlling MGCP endpoint again optimization:

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

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

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

```
ACMEPACKET(configure)# medi a-manager
ACMEPACKET(medi a-manager)#

```

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

```
ACMEPACKET(medi a-manager)# medi a-manager
ACMEPACKET(medi a-manager-confi g)#

```

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

4. **options**—Set the options parameter with the values you want to change by typing a plus sign (+) and then **options** followed by a <Space>. Continuing with the entry, then type the option name with an equal sign followed by the value you want to use for that option. Then press <Enter>. Repeat this process for each of the options you want to change from its default.

Refer to the table in the [How It Works \(724\)](#) to view details about the individual values for this feature and how they work.

```
ACMEPACKET(medi a-manager-confi g)# opti ons +al g-agl ng-max-l oop=150
ACMEPACKET(medi a-manager-confi g)# opti ons +al g-agl ng-max-l oop=10
ACMEPACKET(medi a-manager-confi g)# opti ons +al g-agl ng-cpu-l imi t=40
ACMEPACKET(medi a-manager-confi g)# opti ons +al g-agl ng-l oad-max-l oop=75
ACMEPACKET(medi a-manager-confi g)# opti ons +al g-agl ng-l oad-max-exp=5
```

If you type the option without the “plus” sign, you will overwrite any previously configured options. In order to append the new options to this configuration’s options list, you must prepend the new option with a “plus” sign as shown in the previous example.

5. Save and activate your configuration.

MGCP Configurable Endpoint Removal

You can configure your Net-Net SBC for MGCP configurable endpoint removal, a feature that enables dynamic removal of MGCP endpoints and related session information upon receipt of configured 5xx series permanent error code(s).

In MGCP, a gateway notifies a call agent that a group of endpoints managed by the gateway is going on or out of service by sending a RestartInProgress (RSIP) message. The call agent’s response to the RSIP contains a code reflecting the success or failure of the request:

- Success—Indicated by the return code 200 - transaction executed; restart was completed successfully, and the NotifiedEntity returned is the new notified entity for the endpoint
- Failure—Either indicated by 4xx (transient error) or 5xx (permanent error) return codes; only the 5xx codes are relevant to this feature; restart fail to complete successfully; if there is a NotifiedEntity returned in the response, this new notified entity must be include when the restart is attempted again

For more information about these classes of codes and about individual codes, see RFC 3435, “Media Gateway Control Protocol (MGCP) Version 1.0.”

Without this feature enabled, the Net-Net SBC creates an MGCP session when it receives the RSIP message. This behavior can leave the Net-Net SBC vulnerable to

a DoS attack if the attacker were to willfully misrepresent thousands of endpoints, thereby eventually exhausting the Net-Net SBC's memory capacity.

How It Works

Using the new `rsip-failures` parameter in the MGCP configuration, you can set the 5xx return codes—or series of 5xx codes—that trigger endpoint removal. Because the request creates the endpoint and reserves associated Net-Net SBC resources for it, RSIP are the basis for removal; the Net-Net SBC disregards other requests of the endpoint is not registered first.

The default value for this parameter is a set of ranges: 500-509, 511-519, 522-599. This set return codes excludes numbers 510 (unspecified protocol error) and 520 (endpoint is restarting and the transaction could not be completed) as the errors they reflect are temporary and for which sessions should be established. While return code 521 (redirection) is excluded from the default, it can be included without creating conflict with the existing feature, which removes the MGCP session when a 520 is received. An empty-string entry disables the feature.

ACLI Instructions and Examples

To configure MGCP endpoint removal:

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 session-router path.

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

```
ACMEPACKET(session-router)# mgcp-config
ACMEPACKET(mgcp-config)#
```
4. **`rsip-failures`**—Enter the range of 5xx return codes that you want to trigger MGCP endpoint removal or that will not succeed in creating an MGCP session on the Net-Net SBC. To disable this feature, enter an open quotation mark ("") followed a <Space> and then a close quotation mark ("") for an empty string. The default value is **500-509, 511-519, 522-599**.

```
ACMEPACKET(mgcp-config)# rsip-failures ""
```
5. Save and activate your configuration.

MGCP Port Mapping

The Net-Net SBC supports a gateway masquerading function, which gives an external gateway a unique address representation on the core side using a one-to-one mapping of IP addresses. The Net-Net SBC also supports an MGCP port mapping feature allowing for a mapping of many IP addresses to one IP address on the core gateway side, using unique and dynamically allocated ports for each external gateway. This feature allows you to configure the range from which the port numbers are drawn.

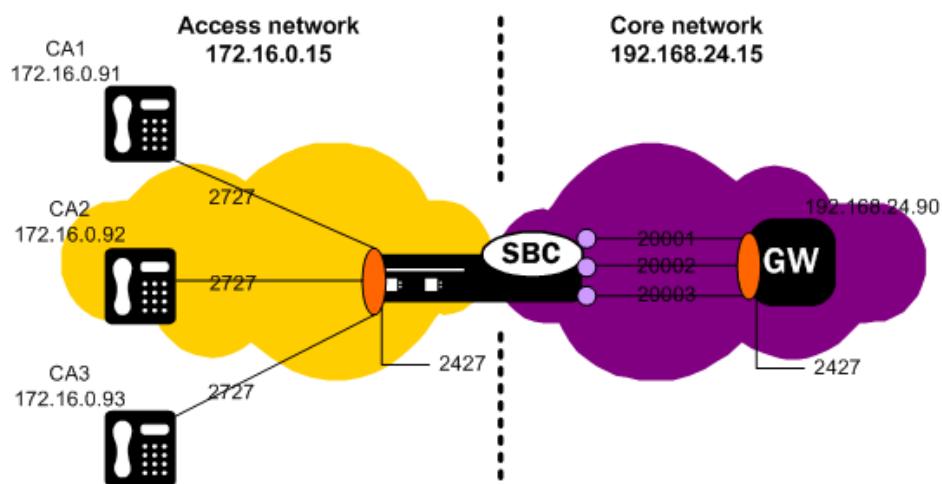
Using the MGCP port mapping feature saves you from having to reserve many IP addresses to use for one-to-one mapping on the Net-Net SBC, and from having to provision many static routes in core network router. This is because when traffic from an endpoint in the public Internet traverses the Net-Net SBC, the Net-Net SBC designates a specific port number to represent the endpoint. The core gateway IP address on the Net-Net SBC remains the same; only the port number changes.

How It Works

For admission control, certain service providers require a unique signaling transport address (IP address and UDP port combination) for each endpoint. However, when deployed in such a network, the Net-Net SBC sits between endpoints and the gateway such that the gateway perceives the Net-Net SBC's IP address as the IP addresses of the individual endpoints. To give endpoints unique transport addresses, the Net-Net SBC allocates a port number for each.

When it receives a session-initiating MGCP message from the access side, the Net-Net SBC refers to a pool of ports you have configured to allocate a unique signaling port. Then, when it forwards an RSIP, the Net-Net SBC replaces the source address and port information with the allocated signaling port address.

The following diagram illustrates how the MGCP port mapping feature works for an access deployment where the MGCP configuration's port mapping feature is enabled.



In this figure, CA1, CA2, and CA3 are call agents in the access network, for which the IP address and port value is 172.16.0.15:2427 (configured in the **private-address** parameter). The configuration for this sample would also show a public gateway address (the **public-gw-address** parameter) of 192.168.24.15. The gateway resides at 192.168.24.90:2427. As you can see, the Net-Net SBC allocates ports 20001 through 20003 for the three calls agents on the access side so that the gateway perceives each as having a unique signalling address rather than all having the signalling of the Net-Net SBC itself. For example, if CA1 were to send an RSIP with a source IP address and port of 172.16.0.91:2727, the Net-Net SBC replaces that information with 192.168.24.15:20001 when it forwards the RSIP to the core network. Likewise, all MGCP requests addressed to the allocated signaling transport address are translated and forwarded with the associated MGCP session.

Availability of Ports in the Pool

Signaling ports are returned to the pool for use with new MGCP sessions when the MGCP session for which it was being used is removed. Removal might occur when a session times out, for example, or when it is explicitly removed with an RSIP request. The newly available port is returned at the bottom of the list of available ports, resulting in a least-used allocation method for signaling port selection.

If there are no ports available for a new registration, the request uses the IP address and port configured for—in this order—the ACLI **public-gw-address** and the **public-gw-port** parameter values.

About MGCP Port Mapping and ACLs

If you use the **show acl** command to view statistics about access control lists (ACLs), you can see information about the ports being used—including port mask information. As a best practice when you are using MGCP port mapping, you want to configure port ranges that match up with bit mask ranges as well as possible.

When you do not enable MGCP port mapping, the Net-Net SBC creates one ACL entry for each MGCP port. With MGCP port mapping enabled, the Net-Net SBC adds one or more entries to support the defined port range, even though ACLs do not support specific port range. To provide this support, the Net-Net SBC masks port ranges that fall on bit boundaries to represent the range as closely as possible.

For example, the entry 192.168.24.15:4096/12 defines 4096 through 8191 as the port range. This is why you should always try to configure port map range that fall on bit boundaries and therefore use the fewest possible ACL entries and to increase accuracy in port range use. The Net-Net SBC ignores MGCP messages received on ports outside the configured range.

If the ACLs added for the port map range fail to cover the MGCP port set in the MGCP configuration, the Net-Net SBC also adds the normal MGCP ACL entry for the MGCP configuration.

Activating Your Configuration with MGCP Port Mapping Changes

The Net-Net SBC supports RTC for MGCP port mapping. However, configuration changes can disrupt service if, among other possible conditions, the range of ports is reduced. For this reason, the Net-Net SBC warns you when you execute the ACLI **activate-config** command and the MGCP port map range has changed.

ACLI Instructions and Examples

To enable this feature, you configure start and end values for the range of port numbers you want the Net-Net SBC to allocate to individual endpoints.

Your configuration must follow these rules or your port range will be invalid. Invalid port ranges prevent this feature from working properly.

The range of ports you enter:

- Must not overlap with configured public call agent port (the ACLI **public-ca-port** parameter) since it might be used for signaling messages that are not associated with MGCP sessions.
- Must not overlap the port range defined in anysteering port configuration using the same IP address as the public gateway address (the ACLI **public-gw-address** parameter).

Overlap prevents the Net-Net SBC from internally processing traffic as it should. While this was also the case prior to when the Net-Net SBC performed any port mapping for MGCP, the ability to define a range of ports increases the potential for overlap.

Note that the ACLI **verify-config** command does not perform overlap checking for your defined port ranges.

To define a port range that enables MGCP port mapping:

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

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

```
2. Type **session-router** and press <Enter> to access the session-router path.

```
ACMEPACKET(configure)# session-router

```

- ACMEPACKET(session-router)#
 3. Type **mgcp-config** and press <Enter>.
 ACMEPACKET(session-router)#
 ACMEPACKET(mgcp-config)#
 4. **port-map-start**—Enter the port number marking the beginning of the range of ports you want to use for MGCP port mapping. The valid range is 1025 through 65535, and the default value is 0. If you leave this parameter set to its default, this feature is disabled.
 5. **port-map-end**—Enter the port number marking the end of the range of ports you want to use for MGCP port mapping. The valid range is 1025 through 65535, and the default value is 0. If you leave this parameter set to its default, this feature is disabled. When you enable MGCP port mapping the **port-map-end** value you set must be greater than the **port-map-start** value.
 6. Save and activate your configuration.

Monitoring Enhancements

The ACCLI **show algd statistics** command has been enhanced to let you see information about:

- Free Map Ports—Number of ports available in the free signaling port pool
- Used Map Ports—Number of signaling ports allocated for MGCP sessions; equal to the number of MGCP sessions when the port mapping feature is used for all core network realms

Note that each MGCP configuration has its own pool of signaling ports, and that this displays shows one set of statistics to count all of them.

ACMEPACKET# **show algd statistics**

16:24:19-117						
State	High	-- Period --	Total	PerMax	Lifetime	-----
Active MGCP Sessions	1	1	0	4	3	1
CA Endpoints	2	2	0	8	6	2
GW Endpoints	2	2	0	8	6	2
Media Sessions	0	0	0	0	0	0
Client Trans	0	0	0	16	12	16
Server Trans	0	0	0	16	12	16
Pending MBCD	0	0	0	0	0	0
MGCP ALGs	1	1	0	1	1	1
Free Map Ports	497	497	0	501	501	501
Used Map Ports	4	4	0	4	3	4
----- Gateway ----- Call Agent -----						
MGCP Transactions		----- Lifetime -----		----- Lifetime -----		
		Recent	Total	PerMax	Recent	Total
Requests received		0	16	12	0	0
Responses sent		0	16	12	0	0
Duplicates received		0	4	4	0	0
Requests sent		0	0	0	0	16
Responses received		0	0	0	0	16
Retransmissions sent		0	0	0	0	0

9 Application Layer Gateway Services

DNS ALG

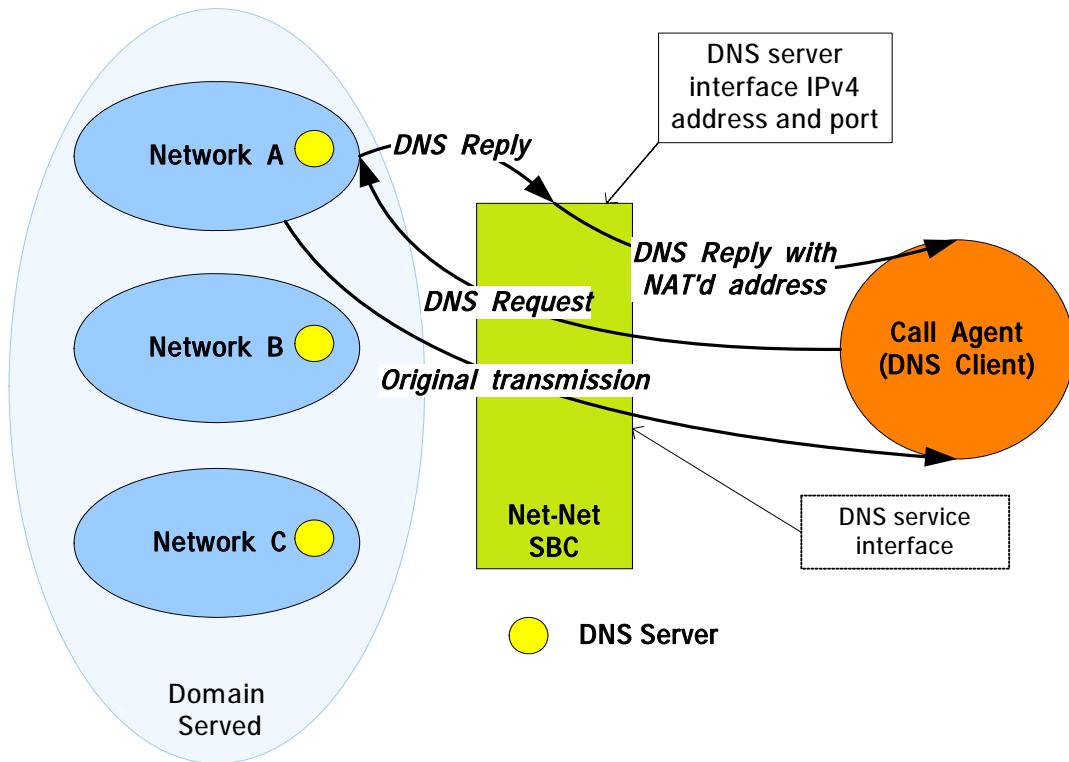
The Net-Net SBC's DNS Application Layer Gateway (ALG) feature provides an application layer gateway for DNS transactions on the Net-Net SBC. With DNS ALG service configured, the Net-Net SBC can support the appearance of multiple DNS servers on one side and a single DNS client on the other.

Overview

DNS ALG service provides an application layer gateway for use with DNS clients. DNS ALG service allows a client to access multiple DNS servers in different networks and provides routing to/from those servers. It also supports flexible address translation of the DNS query/response packets. These functions allow the DNS client to query many different domains from a single DNS server instance on the client side of the network.

The Net-Net SBC's DNS ALG service is commonly used when a DNS client (such as a call agent) needs to authenticate users. In this case, the DNS client that received a message from a certain network would need to authenticate the endpoint in a remote network. Since the DNS client and the sender of the message are on different networks, the Net-Net SBC acts as an intermediary by interoperateing with both.

In the following diagram, the DNS client has received a message from an endpoint in Network A. Since the DNS client is in a different realm, however, the DNS client receives the message after the Net-Net SBC has performed address translation. Then the DNS client initiates a DNS query on the translated address. The Net-Net SBC forwards the DNS request to the DNS server in Network A, using the domain suffix to find the appropriate server. Network A's DNS server returns a response containing its IPv4 address, and then the Net-Net SBC takes that reply and performs a NAT on the private address. The private address is turned into a public one that the DNS client can use to authenticate the endpoint.



Configuring DNS ALG Service

You can access the configuration parameters for DNS ALG service using the Net-Net EMS or the ACCLI. This section tells you how to access and set the values you need depending on the configuration mechanism you choose. It also provides sample configurations for your reference.

Configuring DNS ALG service requires that you carry out two main procedures:

- Setting the name, realm, and DNS service 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 that you are going to use in the main DNS ALG profile and note its name to use in this chapter's configuration process.
2. Configure the server realm that contains the DNS servers and note its name to use in this chapter's configuration process.
3. Determine the domain suffixes for the network where the DNS servers are located so that you can enter them in the domain suffix parameter.
4. Devise the NAT scheme that you want to use when the DNS reply transits the 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-translational
      server-prefix      10.3.0.0/16
      client-prefix       192.168.0.0/16
```

Identity, Realm, and Interface Addresses

To configure the identity, realm, and IPv4 interface addresses for your DNS ALG profile:

1. **description**—Set a name for the DNS ALG profile using any combination of characters entered without spaces. You can also enter any combination with spaces if you enclose the whole value in quotation marks. For example: "DNS ALG service."
2. **client-realm**—Enter the name of the realm from which DNS queries are received. If you do not set this parameter, the DNS ALG service will not work.
3. **client-address-list**—Configure a list of one or more addresses for the DNS server interface. These are the addresses on the Net-Net SBC 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(config)# **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. The default value is 53. The valid range is:
 - Minimum—1025
 - Maximum—65535

10. **transaction-timeout**—Enter the time in seconds that the ALG should keep information to map a DNS server response back to the appropriate client request. After the transaction times out, further response to the original request will be discarded. The default value is **10**. The valid range is:
 - Minimum—0
 - Maximum—999999999
11. **address-translation**—Enter a list of address translations that define the NAT function for the DNS servers.

You can access the NAT parameters for the DNS servers by typing address-translation and pressing enter within the DNS server attributes configuration.

```
ACMEPACKET(dns-config)# server-dns-attributes
ACMEPACKET(server-dns-attributes)# address-translation
```

To configure the NAT, enter two values:

- **server-prefix**: address/prefix that will be returned by the DNS server
- **client-prefix**: address/prefix that to which a response is returned

Each of these is a two-part value:

- IPv4 address
- Number of bits indicating how much of the IPv4 address to match

If you do not specify the number of bits, then all 32 bits of the IPv4 address will be used for matching. If you set the number of bits to 0, then the address will simply be copied.

For example, if you set the server prefix to 10.3.17.2/16 and the client prefix to 192.168.0.0/16, then the Net-Net SBC will return an address of 192.168.17.2 to the DNS client.

```
ACMEPACKET(server-dns-attributes)# address-translation
ACMEPACKET(address-translation)# server-prefix 10.3.17.2/16
ACMEPACKET(address-translation)# client-prefix 192.168.0.0/16
```

DNS Transaction Timeout

To provide resiliency during DNS server failover, you can now enable a transaction timeout for DNS servers. If you have endpoints that are only capable of being configured with a single DNS server, this can allow DNS queries to be sent to the next configured server—even when contacting the 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(medi a-manager)# **medi a-manager**
 ACMEPACKET(medi a-manager-confi g)#
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 your work.

H.248 ALG

You can use the Net-Net SBC to perform the functions of a virtual call agent for H.248 gateways on the access network. On the core side, the H.248 ALG can act as a virtual gateway (one instance for each gateway connecting on the access side) and connect to real call agents. By aggregating signaling and media for many endpoints, it can appear as media gateway controller.

You can configure multiple instances of the virtual call agent and virtual gateway, each with its unique virtual gateway addresses. This way, you can hide topology by keeping external gateways from knowing real call agent addresses.

The H.248 ALG also provides dynamic filters that prevent floods and attacks that use the H.248 protocol. You can define a signaling threshold in bytes per second and then apply the threshold to each gateway individually. If signalling traffic exceeds the threshold, the Net-Net SBC automatically demotes the source IP address.

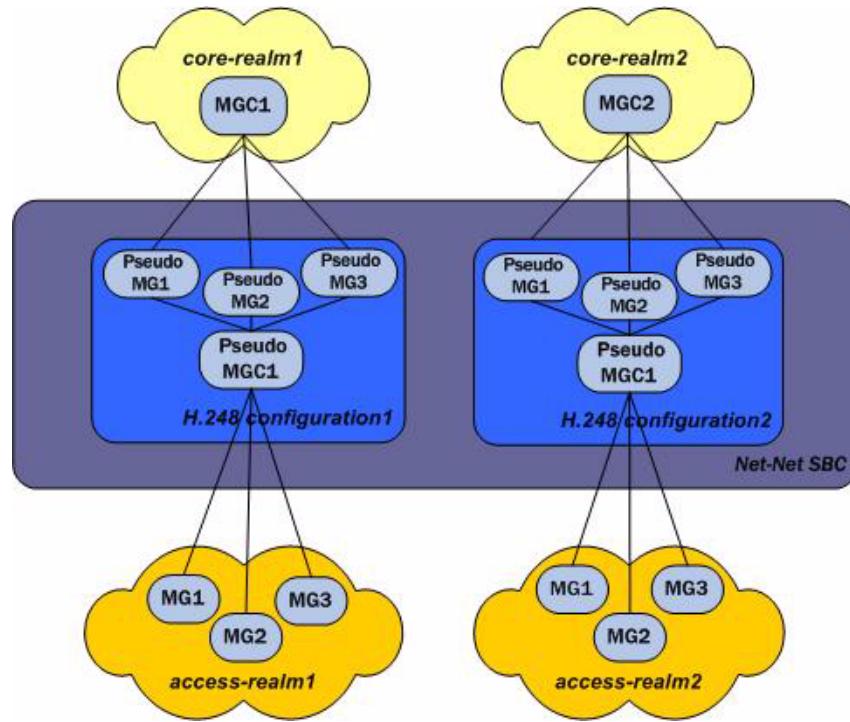
Note that the Net-Net SBC provides DoS protection on the access side.

RTN 1652

Sample Application

In the following diagram, all media gateways send requests to a single IP address and port, which is a virtual (or pseudo) media gateway controller (MGC) on the Net-Net SBC. The Net-Net SBC dynamically creates an IP mapping at the core network for

each registered gateway. To the core side, the Net-Net SBC appears to be many media gateways (MGs) and connect to the real corresponding softswitch.



Gateway Masquerading

The H.248 supports gateway masquerading, where the Net-Net SBC creates an IP mapping at the core network for each registered gateway. To enable gateway masquerading for the H.248 ALG, you need set an ip-address parameter in the h248-mg-config to a value that has an IP address and a mask.

Consider the following configuration:

<pre>h248-mgc-config name ip-address primary-mg</pre>	<pre>virtual MgcOne 192.168.32.251 real MgcOne</pre>
<pre>h248-mg-config mgc-name mgc-ip-address ip-address</pre>	<pre>real MgcOne 192.168.232.211 192.168.232.0/25</pre>

The Net-Net SBC copies all of the host bits—32, minus the netmask from the **ip-address** parameter of the **h248-mg-config**—from the incoming source to the outgoing source. So if the gateway's address arrived at the Net-Net SBC with an incoming value of 192.168.32.10 on the access side, then the outgoing source on the core side would be 192.168.232.10.

Handoff Support

Using an example is the best way to illustrate the Net-Net SBC's handoff support. The example this section discusses uses the following configuration:

```
h248-mgc-config
```

	name	virtualMgcOne
	ip-address	192.168.32.251
	primary-mg	realMgcOne
h248-mgc-config		
	name	virtualMgcTwo
	ip-address	192.168.32.252
	primary-mg	realMgcTwo
h248-mg-config		
	mgc-name	realMgcOne
	mgc-ip-address	192.168.232.211
	ip-address	192.168.232.0/25
h248-mg-config		
	mgc-name	realMgcTwo
	mgc-ip-address	192.168.232.211
	ip-address	192.168.232.0/25

In this example, the gateway on the access side connects to virtualMgcOne. When it receives a registration from the gateway that virtualMgcOne represents, the Net-Net SBC forwards it to the real MGC. It does so using its realMgcOne configuration, which is identified in virtualMgcOne's **primary-mg** setting. From this point, the Net-Net SBC forwards all messages arriving at virtualMgcOne to realMgcOne—and the reverse is also the case.

It is possible, however, that realMgcOne would want to hand off the gateway to realMgcTwo. To do so, it would send a service change message (servicechangeMgcId), containing the IP address and port for realMgcTwo. When it receives the service change message, the Net-Net SBC recognizes that the gateway has been handed off and then selects an H.248 MGC configuration (**h248-mgc-config**) the points to realMgcTwo. As you can see from the sample configuration, it would select virtualMgcTwo. Then the Net-Net SBC forwards the message to the gateway by modifying the servicechangeMgcId so that it contains the IP address for virtualMgcTwo. Now the gateway would register with virtualMgcTwo, and the Net-Net SBC would forward this registration to realMgcTwo.

Licensing

You need to obtain and enable an H.248 license to use the H.248 ALG.

ACLI Instructions and Examples

This section contains information about how to configure:

- Global H.248 ALG functionality
- The H.248 MG configuration
- The H.248 MGC configuration

When you enable the H.248 on your Net-Net SBC, keep in mind the following general steps:

6. For the access side, configure an H.248 MGC configuration (**h248-mgc-config**) to act as the MGC to the gateways on the access side.
7. For the core side, configure an H.248 MG configuration (**h248-mg-config**) to act as the MG on the core side.

8. Create a bridge between the MG and MCG configurations using the primary-mgc parameter (in the h248-mgc-config); this parameter's value is the **mgc-name** of an h248-mgc-config.

Enabling Global H.248 Functionality

To enable global H.248 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 **h248-config** and press <Enter>.

```
ACMEPACKET(session-router)# h248-config
ACMEPACKET(h248-config)#

```
4. **state**—Leave this parameter set to **enabled** (default) if you want to use the Net-Net SBC's support for H.248. Change it to **disabled** if you want to turn off all H.248 functionality.
5. **log-level**—Set the process log level for monitoring all H.248 activity on the Net-Net SBC. The default is **INFO**. Other valid values are: **EMERGENCY**, **CRITICAL**, **MAJOR**, **MINOR**, and **WARNING**. The system will accept **DEBUG**, **INFO**, **TRACE**, and **NOTICE**—but they should only be used in consultation with Acme Packet customer support.

Note that any log level you set here overrides the log level you set in the system configuration's process log level parameter.
6. **red-h248-port**—Enter the port on which H.248 checkpointing messages are sent and received. The default value is 1989. A value of 0 disables the H.248 checkpointing. The minimum value is 1025, and the maximum is 65535.
7. **red-max-trans**—Enter the maximum size of the transaction list, or how many H.248 transactions you want to store in memory at one time. Oldest transactions will be discarded first in the event that the limit is reached. The default value is 10000. The valid range is:
 - Minimum—0
 - Maximum—999999999
8. **red-sync-start-time**—Enter the number of milliseconds before the Net-Net SBC will try to synchronize its signaling state checkpointing. If the active Net-Net SBC is still adequately healthy, this timer will simply reset itself. If for any reason the active Net-Net SBC has become the standby, it will start to checkpoint with the newly active system when this timer expires.

We recommend that you leave this parameter set to its default, 5000. The valid range is:

 - Minimum—0
 - Maximum—999999999
9. **red-sync-comp-time**—Enter the number of milliseconds representing how frequently the standby Net-Net SBC checkpointing with the active Net-Net SBC to obtain the latest H.248 information. The first interval occurs after initial synchronizations of the systems.

We recommend that you leave this parameter set to its default, 1000. The valid range is:

- Minimum—0
- Maximum—999999999

10. Save your work.

Configuring the H.248 MGC

To configure an H.248 MGC:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **h248-config** and press <Enter>.

```
ACMEPACKET(session-router)# h248-config
ACMEPACKET(h248-config)#
```
4. Type **h248-mgc-config** and press <Enter>.

```
ACMEPACKET(h248-config)# h248-mgc-config
ACMEPACKET(h248-mgc-config)#
```
5. **state**—Leave this parameter set to **enabled** (default) if you want to use this H.248 MGC configuration. Change it to **disabled** if you want to turn it off.
6. **realm-id**—Select the realm you want to use for this H.248 MGC configuration. The value for this parameter is the value you entered for the **identifier** parameter in the **realm-config** representing the realm you want to use. This parameter is required, but it has no default value.
7. **ip-address**—Enter the IP address for the Net-Net SBC to use when acting as an MGC on the access side. This parameter is required, but it has no default value.
8. **port**—Enter the port number you want to be associated with the IP address you entered for the **ip-address** parameter. The default value is **2944**. The minimum value for this parameter is **1025**, and the maximum is **65535**.
9. **transport-method**—Enter the transport method you want to use. The default is **UDP**, and only UDP is supported at this time.
10. **encoding**—Enter the encoding method you want to use for this MGC. You can use either **text** (default) or **binary**.
11. **primary-mg**—Enter the **mgc-name** value from the **h248-mg-config** that you want to associate with this MGC configuration. This parameter's value (which is a hostname) creates the bridge between the MGC and the MG configurations. This parameter is required, but it has no default value.
12. **audit-interval**—Enter the time in seconds you want to use as the interval between AuditValue messages. The default for this parameter is 0.
13. **nat-traversal**—Leave this parameter set to **disabled** (default), or change it to **enabled** to signify that gateway being referenced in this configuration is behind NATs.
14. Save your work.

Configuring the H.248 MG

To configure an H.248 MG:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **h248-config** and press <Enter>.

```
ACMEPACKET(session-router)# h248-config
ACMEPACKET(h248-config)#
```
4. Type **h248-mg-config** and press <Enter>.

```
ACMEPACKET(h248-config)# h248-mg-config
ACMEPACKET(h248-mg-config)#
```
5. **mgc-name**—Enter the hostname of the MGC. You use this value to in the **primary-mgc** parameter (in the **h248-mgc-config**) when you want to bridge between an MGC configuration and an MG configuration. This parameter has no default, but it is required.
6. **mgc-ip-address**—Enter the IP address of the real MGC on the core side. This parameter has no default, but it is required.
7. **mgc-port**—Enter the port number you want to be associated with the IP address you entered for the **mgc-ip-address** parameter. The default value is **2944**. The minimum value for this parameter is 1025, and the maximum is 65535.
8. **state**—Leave this parameter set to **enabled** (default) if you want to use this H.248 MG configuration. Change it to **disabled** if you want to turn it off.
9. **realm-id**—Select the realm you want to use for this H.248 MG configuration. The value for this parameter is the value you entered for the **identifier** parameter in the **realm-config** representing the realm you want to use. This parameter is required, but it has no default value.
10. **ip-address**—Enter the local IP address of the MG. By adding a netmask, you can use this parameter to define a masquerading subnet. A value used to set up masquerading would look like this: **192.168.232.0/25**.

This parameter is required, but it has no default value.
11. **state**—Leave this parameter set to **enabled** (default) if you want to use this H.248 MG configuration. Change it to **disabled** if you want to turn it off.
12. **encoding**—Enter the encoding method you want to use for this MGC. You can use either **text** (default) or **binary**.
13. Save your work.

10 Session Routing and Load Balancing

Introduction

This chapter explains how to configure session routing and load balancing for SIP and H.323 services. It contains information about configuring session agents and session agent groups, as well as local policies that can be used for routing SIP or H.323 signals.

Routing Overview

This section provides an overview of routing SIP and H.323 sessions when using the Net-Net SBC. The Net-Net SBC chooses the next hop through the network for each SIP and H.323 session based on information received from routing policies and constraints. Routing policies can be as simple as routing all traffic to a proxy or routing all traffic from one network to another. Routing policies can also be more detailed, using constraints to manage the volume and rate of traffic that can be routed to a specific network. For example, you can manage volume and rate of traffic to enable the Net-Net SBC to load balance and route around softswitch failures.

When a call request arrives at the Net-Net SBC, a decision making process then occurs to determine whether the message is coming from a session agent. If so, the Net-Net SBC checks whether that session agent is authorized to make the call. Local policy is then checked to determine where to send the message on to.

Session Agents, Session Groups, and Local Policy

When you configure session routing for SIP and H.323, you can use session agents, session agent groups and local policies to define routing. (Using session agents and session agent groups is not required.)

- session agent: defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes.
- session agent group (SAG): 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.

Another element of routing is the realm. Realms are used when a Net-Net SBC 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 *Configuring Realms* 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 SBC. The realm identifier value is also needed when you configure session agents and local policies. You can associate a realm with a session agent to identify the realm for sessions coming from or going to the session agent. You also need the realm identifier when you configure local policy to identify the egress realm (realm of the next hop).

About Session Agents

This section describes session agents. A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. Service elements such as gateways, softswitches, and gatekeepers are defined automatically within the Net-Net SBC as session agents. For each session agent, concurrent session capacity and rate attributes can be defined. You can group session agents together into session agent groups and apply allocation strategies to achieve traffic load balancing.

You can assign a media profile to a session agent and indicate whether the transport protocol is SIP or H.323. If the protocol is H.323, you need to indicate whether the session agent is a gateway or a gatekeeper.

You can configure a set of attributes and constraints for each session agent to support the following:

- session access control: Net-Net SBC only accepts requests from configured session agents
- session admission control (concurrent sessions): Net-Net SBC 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 SBC polices the rate of session attempts to send to and receive from a specific session agent.

SIP Session Agents

SIP session agents can include the following:

- softswitches
- SIP proxies
- application servers
- SIP gateways
- SIP endpoints

In addition to functioning as a single logical next hop for a signaling message (for example, where a SIP INVITE is forwarded), session agents can provide information about next or previous hops for packets in a SIP agent, including providing a list of equivalent next hops.

You can use the session agent to describe one or more SIP next or previous hops. Through the configured carriers list, you can identify the preferred carriers to use for traffic coming from the session agent. This set of carriers will be matched against the local policy for requests coming from the session agent. You can also set constraints for specific hops.

Session Agent Status Based on SIP Response

The Net-Net SBC can take session agents out of service based on SIP response codes that you configure, and you can also configure SIP response codes that will keep the session agent in service.

With this feature disabled, the 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 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.

There are two lists you can configure in the session agent configuration to determine status:

- In-service list—Set in the ACLI **ping-in-service-response-codes** parameter, this list defines the response codes that keep a session agent in service when they appear in its response to the 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 ACLI **out-service-response-codes** parameter, this list defines the response codes that take a session agent out of service when they appear in its response to the Net-Net SBC's ping request or any dialog-creating 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.

SIP Session Agent Continuous Ping

You can configure the Net-Net SBC to use either a keep-alive or continuous method for pinging SIP session agents to determine their health—i.e., whether or not the Net-Net SBC should route requests to them. To summarize the two methods:

- **keep-alive**—The Net-Net SBC sends a ping message of a type you configure to the session agent in the absence of regular traffic. Available in Release C5.1.0 and in earlier releases.
- **continuous**—The Net-Net SBC sends a ping message regardless of traffic state (regular or irregular); the Net-Net SBC regularly sends a ping sent based on the configured ping interval timer. Available in Release C5.1.1p6 and in later releases.

How It Works

By sending ping messages, the Net-Net SBC monitors session agents' health and can determine whether or not to take a session out of service (OOS), leave it in service, or bring it back into service after being OOS.

When you set it to use the keep-alive mode of pinging (available in Release C5.1.0 and before), the Net-Net SBC starts sending a configured ping message to a session agent when traffic for that session agent has become irregular. The Net-Net SBC only sends the ping if there are no SIP transactions with a session agent over a configurable period of time, to which the session agent's response can have one of the following results:

- **Successful response**—A successful response is either any SIP response code or any response code not found in the **out-service-response-codes** parameter; these leave the session agent in service. In addition, any successful response or any response in the **ping-in-service-response-codes** parameter can bring a session agent from OOS to in-service status.
- **Unsuccessful response**—An unsuccessful response is any SIP response code configured in the **out-service-response-codes** parameter and takes the session agent sending it OOS. Because this parameter is blank by default, the Net-Net SBC considers any SIP response code successful.
- **Transaction timeout**—A transaction timeout happens when the session agent fails to send a response to the Net-Net SBC's request, resulting in the session agent's being taken OOS.

Despite the fact that the keep-alive ping mode is a powerful tool for monitoring session agents' health, you might want to use the continuous ping method if you are concerned about the Net-Net SBC not distinguishing between unsuccessful responses from next-hop session agents and ones from devices downstream from the next-hop session agent. For example, if a SIP hop beyond the session agent responds with a 503 Service Unavailable, the Net-Net SBC does not detect whether a session agent or the device beyond it generated the response.

When you use the continuous ping method, only the next-hop session agent responds—preventing the request from being sent to downstream devices. The Net-Net SBC also sends the ping in regular traffic conditions when in continuous ping mode, so it is certain the response comes from the next hop associated with the session agent. And in continuous ping mode, only entries for the **ping-out-service-response-codes** parameter and transaction timeouts bring session agents OOS.

ACLI Instructions and Examples

You can set the ping mode in the session agent or session constraints configuration. For backward compatibility, the default for the **ping-send-mode** parameter is keep-alive, or the functionality available in Release C5.1.0 and in earlier releases.

To configure the ping mode for a session agent:

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

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

```

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

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

```

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

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

```

If you are adding rate constraints to an existing configuration, then you will need to select the configuration you want to edit.

4. **ping-send-mode**—If to want to use continuous ping mode to send ping messages to session agents in regular traffic conditions, set this parameter to **continuous**. If you want to use the keep-alive mode, leave this parameter set to **keep-alive** (default).

5. Save and activate your configuration.

To configure the ping mode for the session constraints:

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

```
ACMEPACKET# config 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. **ping-send-mode**—If to want to use continuous ping mode to send ping messages to session agents in regular traffic conditions, set this parameter to **continuous**. If you want to use the keep-alive mode, leave this parameter set to **keep-alive** (default).

5. Save and activate your configuration.

H.323 Session Agents

H.323 session agents can include the following:

- Gatekeepers
- Gateways
- MCUs

Overlapping H.323 Session Agent IP Address and Port

You can now configure H.323 session agents to use overlapping IP addresses.

H.323 session agents continue are identified by their hostname when used in referencing configuration parameters—such as local policy next hops and session agent group destinations. This is why the hostname must be unique. However, when the Net-Net SBC selects a session agent to use, it chooses the appropriate realm and H.323 stack based on the hostname. This is the case even if there are other session agents with the same IP address and port. Likewise, incoming calls are matched to the session agent based on the incoming realm.

There are no specific parameters to configure in order to enable this feature. For it to work properly, however, each H.323 session agent must be configured with a unique hostname (still the primary index). Otherwise, session agents with non-unique hostnames will overwrite one another.

To create overlapping H.323 session agents, you give each of them a unique hostname, which only serves to identify each individually. The Net-Net SBC subsequently uses this label as the next hop destination in relevant local policy route entries.

Managing Session Agent Traffic

The Net-Net SBC 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 SBC to suspend the routing of traffic to a session agent, along with the criteria for restoring the route.

Constraint Condition	SIP Criteria	H.323 Criteria	Action	Criteria for Resuming
Maximum sessions exceeded	Maximum concurrent SIP sessions exceeded.	Maximum concurrent H.323 sessions exceeded. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum sessions value.
Maximum outbound sessions exceeded	Maximum concurrent outbound SIP sessions exceeded.	Maximum concurrent outbound H.323 sessions exceeded. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	Session agent is declared in constraint violation state.	Concurrent sessions drop below the maximum outbound sessions value.

Constraint Condition	SIP Criteria	H.323 Criteria	Action	Criteria for Resuming
Maximum burst rate exceeded	Maximum burst rate exceeded in current window.	Maximum burst rate exceeded in current window. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	Session agent is declared in constraint violation state.	Burst rate in subsequent window drops below maximum burst rate.
Maximum sustained rate exceeded	Maximum sustained rate exceeded in current window.	Maximum burst rate exceeded in current window. If the session agent is a gatekeeper and gatekeeper routed mode is not used, this constraint is an aggregate of all the destination gateways. Only maximum outbound sessions are measured.	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.	<ul style="list-style-type: none"> Response timer expires. The default is T301=4 seconds. Connect timer expires. The default is T303=32 seconds. <p>If the session agent is a peer gatekeeper, the LRQ response time is used to determine availability. The RAS response timer is 4 seconds.</p>	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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC routes traffic to the session agent with the lowest sustained session rate, based on observed sustained session rate.

You apply allocation strategies to select which of the session agents that belong to the group should be used. For example, if you apply the Hunt strategy session agents are selected in the order in which they are listed.

SIP Session Agent Group Recursion

You can configure a SIP session agent group (SAG) to try all of its session agents rather than to the next-best local policy match if the first session agent in the SAG fails.

With this feature disabled, the 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 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.

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.

About Local Policy

This section explains the role of local policy. Local policy lets you indicate where session requests, such as SIP INVITES, should be routed and/or forwarded. You use a local policy to set a preference for selecting one route over another. The local policy contains the following information that affects the routing of the SIP and H.323 signaling messages:

- information in the From header

Information in the message's From header is matched against the entries in the local policy's from address parameter to determine if the local policy applies.

- list of configured realms

This list identifies from what realm traffic is coming and is used for routing by ingress realm. The source realms identified in the list must correspond to the valid realm IDs you have already configured

- local policy attributes

The attributes serve as an expression of preference, a means of selecting one route over another. They contain information such as the next signaling address to use (next hop) or whether you want to select the next hop by codec, the realm of the next hop, and the application protocol to use when sending a message to the next hop. You can also use the attributes to filter specific types of traffic.

Routing Calls by Matching Digits

Local policy routing of a call can be based on matching a sequence of digits against what is defined in the local policy. This sequence refers to the first digits in the (phone) number, matching left to right.

The following examples show how the Net-Net SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC selects the first local policy based on the From address value of 12.

SIP and H.323 Interworking

You need to configure local policies, including the requisite local policy attributes, to use the H.323<→SIP interworking (IWF). Flow progression in H.323<→SIP traffic depends heavily on the local policies configured for the Net-Net SBC, which determine what protocol is used on the egress side of a session.

You set the application protocol (an local policy attribute option) to instruct the Net-Net SBC to interwork the protocol of an ingress message into a different protocol (H.323<→SIP or SIP→H.323) upon its egress to the next hop.

For example, if the application protocol is set to SIP, an inbound H.323 message will be interworked to SIP as it is sent to the next hop. An inbound SIP message would pass to the next hop unaffected. If the application protocol is set to H323, an inbound SIP message will be interworked to H.323 before being sent to the next hop.

See *Configuring SIP and H.323 IWF Signaling* for more information.

Route Preference

The Net-Net SBC builds a list of possible routes based on the source realm and the From-address (From-URI) and To-address (Request-URI), which forms a subset from which preference then decides. Any local policy routes currently outside of the configured time/day are not used, if time/day are set. Also, any local policy routes not on the list of carriers (if carriers is set and the requests has a Carrier header) are not used.

Note: Source realm is used in the local policy lookup process, but it is not used in route preference calculations.

The Net-Net SBC applies preference to configured local policies in the following order:

1. Cost (cost in local policy attributes) is always given preference.
2. Matching media codec (media profiles option in local policy attributes).
3. Longest matching To address (to address list in local policy).
4. Shortest matching To address (to address list in local policy).
5. Longest matching From address (from address list in local policy).
6. Shortest matching From address (from address list in local policy).
7. Narrowest/strictest day of week specification (days of week option in local policy attributes).
8. Narrowest/strictest time of day specification (start time and end time options in local policy attributes).
9. Wildcard matches (use of an asterisk as a wildcard value for the from address and to address lists in local policy).

10. Wild card matches are given the least preference. A prefix value of 6 is given a higher preference than a prefix value of * even though both prefix values are, in theory, the same length.

DTMF-Style URI Routing

The 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.

SIP Routing

This section describes SIP session routing. When routing SIP call requests, the Net-Net SBC 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 SBC 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 SBC 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 SBC to use all of the routes available to it.

A Net-Net SBC configured for H.323 architectures will have access to all of the routes it looks up by default.

About Loose Routing

According to RFC 3261, a proxy is loose routing if it follows the procedures defined in the specification for processing of the Route header field. These procedures separate the destination of the request (present in the Request-URI) from the set of proxies that need to be visited along the way (present in the Route header field).

When the SIP NAT's route home proxy field is set to enabled, the Net-Net SBC 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 SBC 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 SBC through a network other than the home realm. The Net-Net SBC uses egress realms for signaling purposes (when matching flows). When a packet arrives at the Net-Net SBC with a destination address that does not match any defined session agents, the Net-Net SBC uses the address associated with the realm that is, in turn, associated with the SIP configuration's egress realm ID, as the outgoing network. With the use of the egress realm ID, it is possible to define a default route for SIP requests addressed to destinations outside the home realm. If no egress realm is defined, the home realm (default ingress realm) is used as the default egress realm.

With session agent egress realm configured, the Net-Net SBC adds a default egress realm to the session agent to identify the signaling interface used for ping requests. The Net-Net SBC also uses the default egress realm when the normal routing request does not yield an egress realm—for example, when a local policy does not specify the next hop's realm.

When you configure session agents, you can define them without realms or you can wildcard the realm value. These are global session agents, and multiple signaling interfaces can reach them. Then, when you use session agent pinging, the Net-Net SBC sends out ping requests using the signaling interface of the default egress realm defined in the global SIP configuration. The global session agents in certain environments can cause problems when multiple global session agents residing in multiple networks, some of which might not be reachable using the default SIP interface egress realm.

The Net-Net SBC uses the session agent egress realm for ping messages even when the session agent has a realm defined. For normal request routing, the Net-Net SBC uses the egress realm for global session agents when local policies or SIP-NAT bridge configurations do not point to an egress realm.

Ping Message Egress Realm Precedence

For ping messages, the egress realm precedence occurs in the following way (in order of precedence):

- Egress realm identified for the session agent.
- Session agent realm (set in the realm-id parameter) or the wildcarded value

- Global SIP configuration egress realm, when configured in the egress-realm parameter
- Global SIP configuration home realm

Normal Request Egress Realm Precedence

For normal request routing, the egress realm precedence occurs in the following way (in order of precedence):

- Egress SIP-NAT realm, when the **route-home-proxy** parameter is set to **forced** and no local policy match is found
- Matching local policy realm, when configured in the local policy attributes
- Session agent realm (set in the **realm-id** parameter) or the wildcarded value
- Session agent egress realm, when configured in the **egress-realm-id** parameter
- Global SIP configuration egress realm, when configured in the **egress-realm** parameter
- Global SIP configuration home realm

ACLI Instructions and Examples

Configuring a session agent egress realm is optional.

To configure a session agent egress realm:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

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

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

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

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

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

4. **egress-realm-id**—Enter the name of the realm you want defined as the default egress realm used for ping messages. The Net-Net SBC will also use this realm when it cannot determine the egress realm from normal routing. There is no default value for this parameter.
5. Save and activate your configuration.

About SIP Redirect

SIP redirect involves proxy redirect and tunnel redirect.

Proxy Redirect

You can configure the SIP proxy mode to define how the SIP proxy will forward requests coming from the session agent. This value is used if the session agent's proxy mode has no value (is empty).

Tunnel Redirect

You can use tunnel redirect when requests are routed to a server behind a SIP NAT that sends redirect responses with addresses that should not be modified by the SIP NAT function. For example, a provider might wish to redirect certain calls (like 911) to a gateway that is local to a the UA that sent the request. Since the gateway address

is local to the realm of the UA, it should not be modified by the SIP NAT of the server's realm. Note that the server must have a session agent configured with the redirect-action field set to the proxy option in order to cause the redirect response to be sent back to the UA.

SIP Method Matching and To Header Use for Local Policies

SIP Methods 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.

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, PRACK, OPTIONS, INFO, SUBSCRIBE, NOTIFY, REFER, UPDATE, 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 this methods list configured. 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# config terminal
ACMEPACKET(config)#
```
2. Type **session-router** and press <Enter>.

```
ACMEPACKET(config)# 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(localhost->#) # 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.
6. Save and activate your configuration.

H.323 Routing

This section describes H.323 routing.

Egress Stack Selection

Static Stack Selection

Egress stack selection includes static stack selection and policy-based stack selection

In static stack selection, the outgoing stack is determined through the establishment of associated stacks in the h323 stack.

The incoming stack (configured in the h323 stack) uses its associated stack value to determine the associated outgoing stack. The associated stack value corresponds to the name of an h323 stack. This type of selection is referred to as *static* because the incoming stack always uses the stack specified in the associated stack as the outgoing stack; no other stacks are considered.

Policy-Based Stack Selection

The Net-Net SBC performs dynamic, policy-based stack selection when an H.323 call arrives at the Net-Net SBC and a configured associated outgoing stack cannot be found.

For policy-based stack selection, the Net-Net SBC refers to local policies that contain address information that corresponds to incoming traffic. This information is contained in the local policy's To address and From address fields. For the source, this information is matched with the Q.931 calling party number; if there is no calling party number, the H.323 source address is used. For the destination, this information is matched with the called party number; if there is no called party number, then the H.323 destination address is used.

After a local policy corresponding to the incoming traffic has been found, the Net-Net SBC looks at the next hop value (a local policy attribute) and selects a local policy for the basis of stack selection. If the local policy look-up yields multiple local policies with the same next hop values, but with different cost values, the local policy with the lowest cost value is selected.

If a realm is not defined in the local policy, the next hop address is then matched against the address prefix values for the realms that are configured for the system. Thus, the Net-Net SBC discovers the realm for this traffic. Using this realm information, the Net-Net SBC performs stack selection. It uses the first configured H.323 stack in the Net-Net SBC's configuration that has a realm ID value matching the identifier field of the realm with the appropriate address prefix.

In the following example, the local policy matching yields a local policy with a next hop value of 169.125.4.1, which corresponds to RealmB. The outgoing stack selected is Stack 3 because it is the first stack to have been configured with RealmB as the realm ID.

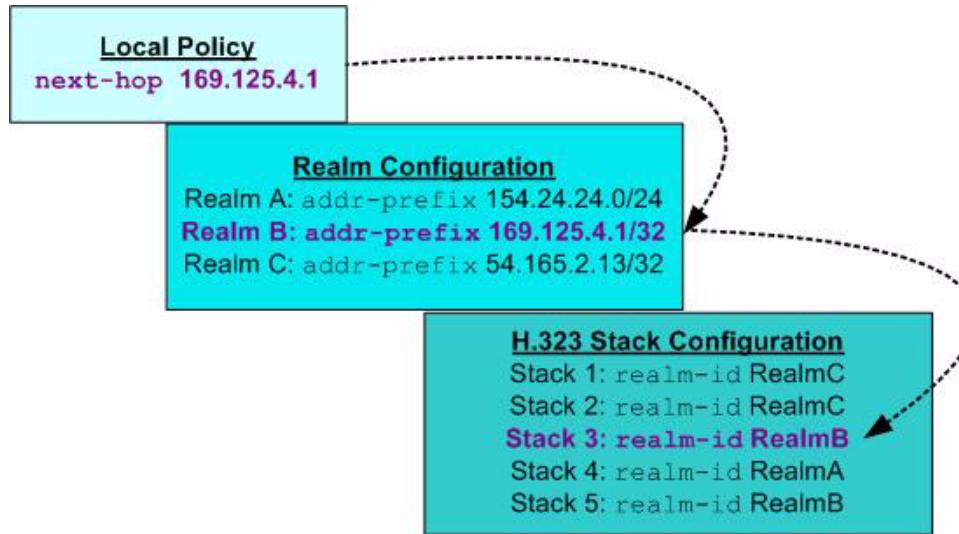


Figure 10-1: Policy-Based Stack Selection

Registration Caching

The Net-Net SBC can cache and proxy an H.225 RegistrationRequest (RRQ) between an H.323 endpoint and a gatekeeper. Registration caching serves two functions:

- It allows aggregation of RRQs sent to a gatekeeper stack and proxies those requests through the gateway stack. If the external gatekeeper associated with the gatekeeper stack supports additive registration, the requests will be consolidated. Furthermore, if the gatekeeper supports additive registration, the Net-Net SBC will register in an additive manner, meaning that will send additive RRQs.
- It allows the gatekeeper stack to use the registration information to route calls from other realms to endpoints in its realms.

To perform registration caching, the Net-Net SBC must be configured with at least two stacks. One of these stacks will receive registrations (*gatekeeper stack*), and one stack will proxy registrations (*gateway stack*). The Net-Net SBC caches all successful registrations and uses the cache to route calls to the endpoints.

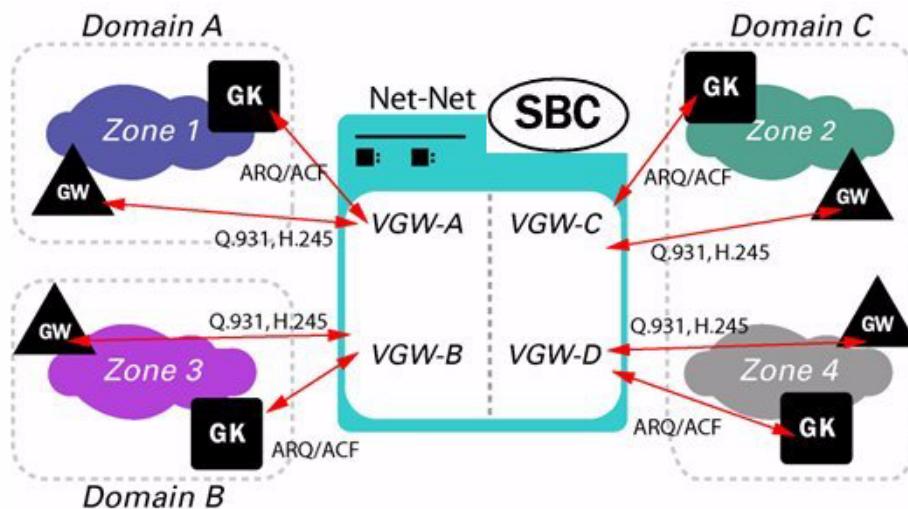
Gatekeeper Provided Routes

Gatekeeper provided routes includes back-to-back gateways, back-to-back gatekeeper and gateway, and interworking gatekeeper/gateway.

Back-to-Back Gateway

When the Net-Net SBC is functioning as a back-to-back gateway (B2BGW), it appears as multiple H.323 gateways to multiple networks. Each Net-Net SBC virtual gateway discovers and registers with a gatekeeper in its respective domain. Each gateway relies on its gatekeeper for admission and location services through the ARQ/ACF exchange. H.225 call control and H.245 messages are exchanged directly with the terminating gateway or gatekeeper. Routing policies are used to associate one virtual gateway with another.

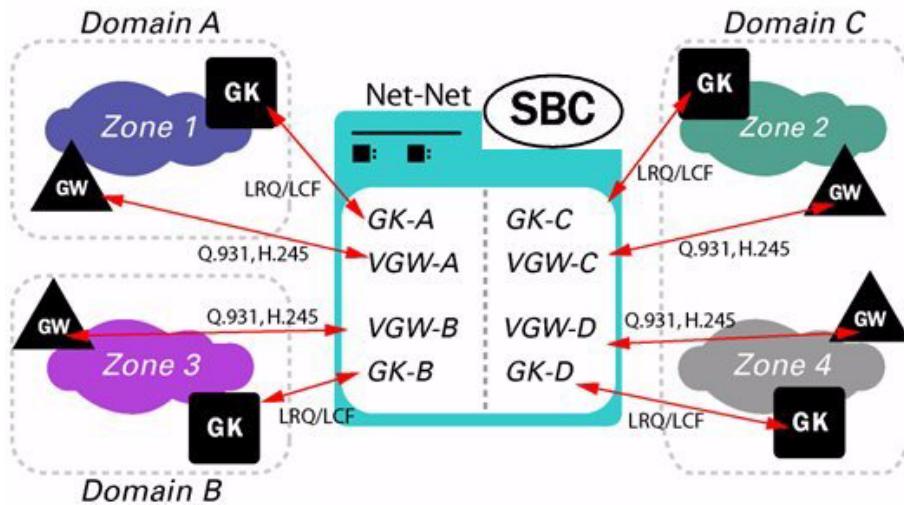
The following diagram illustrates the back-to-back gateway.



Back-to-Back Gatekeeper and Gateway

For peering connections where both networks use inter-domain gatekeeper signaling, the Net-Net SBC is configured as a back-to-back gatekeeper proxy and gateway mode of operation. The Net-Net SBC responds and issues LRQs and LCFs/LRJs acting as a routed gatekeeper. Peered gatekeepers send LRQ to the RAS address of one of the Net-Net SBC's virtual gatekeepers and it responds by providing its call signaling address that performs the gateway functions. Routing policies are used to determine the egress virtual gatekeeper that then exchanges LRG/LCF to determine the call signaling address of the terminating gateway.

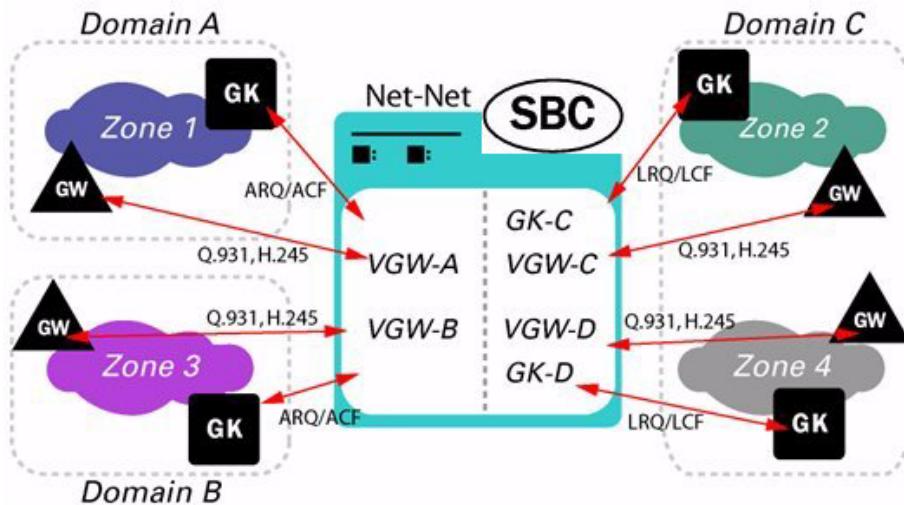
The following diagram illustrates the back-to-back gatekeeper and gateway.



Interworking Gatekeeper/Gateway

In the interworking gatekeeper/gateway signaling mode of operation, the Net-Net SBC interworks between the other two modes; presenting a routed gatekeeper interface to one zone and a gateway to the other.

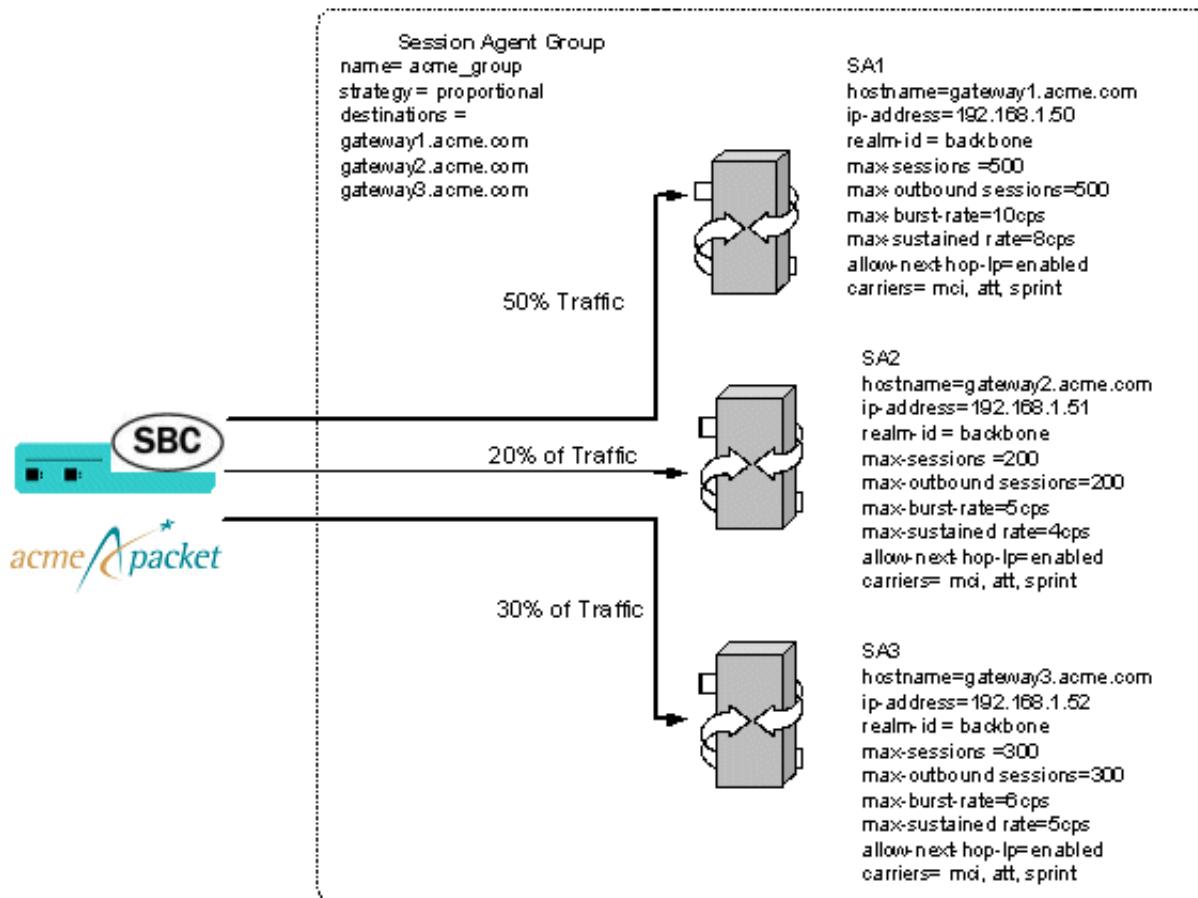
The following diagram illustrates the interworking gatekeeper/gateway.



Load Balancing

This section describes Net-Net SBC 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 SBC.

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

ACLI 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.

Configuring Session Agents

To configure session agents:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# session-router
3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
4. **host-name**—Enter the name of the host associated with the session agent in either hostname or FQDN format, or as an IP address.

If you enter the host name as an IP address, you do not have to enter an IP address in the optional IP address parameter. If you enter the host name in FQDN format, and you want to specify an IP address, enter it in the optional IP address parameter. Otherwise you can leave the IP address parameter blank to allow a DNS query to resolve the host name.

If the initial DNS query for the session agent fails to get back any addresses, the session agent is put out-of-service. When session agent is pinged, the DNS query is repeated. The ping message is not sent until the DNS query gets back

one or more IP addresses. After the query receives some addresses, the ping message is sent. The session agent remains out of service until one of the addresses responds.

Note: The value you enter here must be unique to this session agent. No two session agents can have the same hostname.

The hostnames established in the session agent populate the corresponding fields in other elements.

5. **ip-address**—*Optional*. Enter the IP address for the hostname you entered in FQDN format if you want to specify the IP address. Otherwise, you can leave this parameter blank to allow a DNS query to resolve the host name.
6. **port**—Enter the number of the port associated with this session agent. Available values include:
 - zero (0)—If you enter zero (0), the Net-Net SBC 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 SBC will initiate communication on that port of the session agent.

7. **state**—Enable or disable the session agent by configuring the state. By default, the session agent is **enabled**.
 - enabled | disabled
8. **app-protocol**—Enter the protocol on which you want to send the message. The default value is **SIP**. Available values are:
 - SIP | H.323
9. **app-type**—If configuring H.323, indicate whether the application type is a gateway or a gatekeeper. Available values include:
 - **H.323-GW**—gateway
 - **H.323-GK**—gatekeeper
10. **transport-method**—Indicate the IP protocol to use (transport method) to communicate with the session agent. **UDP** is the default value. The following protocols are supported:
 - **UDP**—Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
 - **UDP+TCP**—Allows an initial transport method of UDP, followed by a subsequent transport method of TCP if and when a failure or timeout occurs in response to a UDP INVITE. If this transport method is selected, INVITES are always sent through UDP as long as a response is received.
 - **DynamicTCP**—dTCP indicates that dynamic TCP connections are the transport method for this session agent. A new connection must be established for each session originating from the session agent. This connection is torn down at the end of a session.
 - **StaticTCP**—sTCP indicates that static TCP connections are the transport method for this session agent. Once a connection is established, it remains and is not torn down.

11. **realm-id**—*Optional.* Indicate the ID of the realm in which the session agent resides.

The realm ID identifies the realm for sessions coming from or going to this session agent. For requests coming from this session agent, the realm ID identifies the ingress realm. For requests being sent to this session agent, the realm ID identifies the egress realm. In a Net-Net SBC, when the ingress and egress realms are different, the media flows must be *steered* between the realms.

- no value: the egress realm is used unless the local policy dictates otherwise
- asterisk (*): keep the egress realm based on the Request URI

Note: The realm ID you enter here must match the valid identifier value entered when you configured the realm.

12. **description**—*Optional.* Enter a descriptive name for this session agent.
13. **carriers**—*Optional.* Add the carriers list to restrict the set of carriers used for sessions originating from this session agent.

Carrier names are arbitrary names that can represent specific service providers or traditional PSTN telephone service providers (for sessions delivered to gateways). They are global in scope, especially if they are exchanged in TRIP. Therefore, the definition of these carriers is beyond the scope of this documentation.

You could create a list using carrier codes already defined in the North American Numbering Plan (NANP); or those defined by the local telephone number or carrier naming authority in another country.

Note: If this list is empty, any carrier is allowed. If it is not empty, only local policies that reference one or more of the carriers in this list will be applied to requests coming from this session agent.

14. **allow-next-hop-lp**—Indicate whether this session agent can be used as a next hop in the local policy.

If you retain the default value of **enabled**, the session agent can be used as the next hop for the local policy. Valid values are:

- enabled | disabled

15. **constraints**—Enable this parameter to indicate that the individual constraints you configure in the next step are applied to the sessions sent to the session agent. Retain the default value of **disabled** if you do not want to apply the individual constraints. Valid values are:

- enabled | disabled

Note: In general, session control constraints are used for SAGs or SIP proxies outside or at the edge of a network.

16. Enter values for the individual constraints you want applied to the sessions sent to this session agent. The following table lists the available constraints along with a brief description and available values.

Constraint	Description
maximum sessions	<p>Maximum number of sessions (inbound and outbound) allowed by the session agent. The range of values is:</p> <ul style="list-style-type: none"> minimum: zero (0) is the default value and means there is no limit maximum: $2^{32} - 1$
maximum outbound sessions	<p>Maximum number of simultaneous outbound sessions (outbound from the Net-Net SBC) that are allowed from the session agent. The range of values is:</p> <ul style="list-style-type: none"> minimum: zero (0) is the default value and means there is no limit maximum: $2^{32} - 1$ <p>The value you enter here cannot be larger than the maximum sessions value.</p>
maximum burst rate	<p>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.</p> <p>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.</p> <p>The range of values is:</p> <ul style="list-style-type: none"> minimum: zero (0) session invitations per second maximum: $2^{32} - 1$ session invitations per second <p>Zero is the default value.</p>

Constraint	Description
maximum sustain rate	<p>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.</p>
	<p>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. 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.</p>
time to resume	<p>The range of values is:</p> <ul style="list-style-type: none"> • minimum: zero (0) invitations per second • maximum: $2^{32} - 1$ invitations per second <p>Zero is the default value.</p> <p>The value you set here must be larger than the value you set for the maximum burst rate constraint.</p>
time to resume (ttr) no response	<p>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.</p> <p>The range of values is:</p> <ul style="list-style-type: none"> • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds <p>Default is zero.</p> <p>The value you enter here must be larger than the value you enter for the time to resume constraint.</p>

Constraint	Description
in service period	<p>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.</p>
	<p>The range of values is:</p>
	<ul style="list-style-type: none"> • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds
	<p>Default is zero.</p>
burst rate window	<p>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.</p>
	<p>The range of values is:</p>
	<ul style="list-style-type: none"> • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds
	<p>Zero is the default value.</p>
	<p>The value you set here must be smaller than the value you set for the maximum burst rate constraint.</p>
sustain rate window	<p>Sustained window period (in seconds) that is used to measure the sustained rate. Refer to the maximum sustain rate information.</p>
	<p>The range of values is:</p>
	<ul style="list-style-type: none"> • minimum: zero (0) seconds • maximum: $2^{32} - 1$ seconds
	<p>Zero is the default value.</p>
	<p>The value you set here must be larger than the value you set for the maximum sustain rate constraint.</p>

17. **req-uri-carrier-mode**—*SIP only*. Set whether you want the selected carrier (determined by a value in the local policy) added to the outgoing message by configuring the request uri carrier mode parameter.

You can set this parameter to let the system perform simple digit translation on calls sent to gateways. A 3-digit prefix is inserted in front of the telephone number (the Request-URI) that the gateway will use to select a trunk group. Most often, the Net-Net SBC needs to insert the carrier code into the signaling message that it sends on.

The default value is **none**. The following lists the available modes.

- **none**—Carrier information will not be added to the outgoing message.
- **uri-param**—Adds a parameter to the Request-URI. For example, cic-XXX.
- **prefix**—Adds the carrier code as a prefix to the telephone number in the Request-URI (in the same manner as PSTN).

18. **proxy-mode**—*SIP only*. Indicate the proxy mode to use when a SIP request arrives from this session agent.

If this field is empty (upon initial runtime or upgrade), its value is set to the value of the SIP configuration's proxy mode by default. If no proxy mode value was entered for the SIP configuration, the default for this field is **proxy**.

The following are valid proxy modes:

- **proxy**—If the Net-Net SBC 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 SBC is a Session Director, the system behaves as a B2BUA when forwarding the request.
- **redirect**—The system sends a SIP 3xx reDIRECT response with contacts (found in the local policy) to the previous hop.

19. **redirect-action**—*SIP only*. Indicate the action you want the SIP proxy to take when it receives a Redirect (3XX) response from the session agent.

If the response comes from a session agent and this field is empty (upon initial runtime or upgrade), the redirect action will be **recurse**. If no session agent is found (for example, if a message comes from an anonymous user agent), the redirect action is set to **proxy**. If the Redirect (3xx) response does not have any Contact header, the response will be sent back to the previous hop.

The following table lists the available proxy actions along with a brief description

- **proxy**—The SIP proxy passes the response back to the previous hop; based on the 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.

20. **loose-routing**—*SIP only*. Enable this parameter if you want to use loose routing (as opposed to strict routing). The default is **enabled**. Valid values are:

- enabled | disabled

When the SIP NAT route home proxy parameter is enabled, the Net-Net SBC looks for a session agent that matches the home proxy address and checks the loose routing value. If loose routing is enabled, a Route header is included in the outgoing request in accordance with RFC 3261. If loose routing is disabled, the Route header is not included in the outgoing request (in accordance with strict routing procedures defined in RFC 2543).

The loose routing value is also checked when the local policy's next hop value matches a session agent. If loose routing is set to enabled, the outgoing request retains the original Request-URI and Route header with the next hop address.

21. **send-media-session**—*SIP only*. Enable this parameter if you want to include a media session description (for example, SDP) in the INVITE or REINVITE message sent by the Net-Net SBC. Setting this field to **disabled** prevents the Net-Net SBC from establishing flows for that INVITE message.

The default is **enabled**. Valid values are:

- enabled | disabled

Note: Only set send media session to disabled for a session agent that always redirects requests. It returns an error or 3xx response instead of forwarding an INVITE message.

22. **response-map**—*Optional and for SIP only.* Enter the name of the response map to use for this session agent. The mappings in each SIP response map is associated with a corresponding session agent. You can also configure this value for individual SIP interfaces.

23. **ping-method**—*SIP only.* Indicate the SIP message/method to use to ping a session agent. The ping confirms whether the session agent is in service. If this field is left empty, no session agent will be pinged.

Setting this field value to the OPTIONS method might produce a lengthy response from certain session agents and could potentially cause performance degradation on your Net-Net SBC.

24. **ping-interval**—*SIP only.* Indicate how often you want to ping a session agent by configuring the ping interval parameter. Enter the number of seconds you want the Net-Net SBC to wait between pings to this session agent. The default value is **0**. The valid range is:

- Minimum: 0
- Maximum: 999999999

The Net-Net SBC only sends the ping if no SIP transactions (have occurred to/from the session agent within the time period you enter here).

25. **trunk-group**—Enter up to 500 trunk groups to use with this single session agent. Because of the high number of trunk groups you can enter, the ACLI provides enhanced editing mechanisms for this parameter:

- You use a plus sign (+) to add single or multiple trunk groups to the session agent's list.

When you add a single trunk group, simply use the plus sign (+) in front of the trunk group name and context. Do not use a <Space> between the plus sign and the trunk group name and context.

For example, you might have already configured a list of trunk groups with the following entries: **tgrpA:contextA**, **tgrpB:contextB**, and **tgrpC:contextC**. To add **tgrp1:context1**, you would make the following entry:

```
ACMEPACKET(session-agent)# trunk-group +tgrp1:context1
```

Your list would then contain all four trunk groups.

When you add multiple trunk groups, simply enclose your entry in quotation marks ("") or in parentheses (()). While you put spaces between the trunk group name and context entries, you do not use spaces with the plus sign, parentheses or quotation marks.

```
ACMEPACKET(session-agent)# trunk-group +"tgrp1:context1"
tgrp2:context2 tgrp3:context3"
```

- You use a minus sign (-) to delete single or multiple trunk groups from the session agent's list.

When you remove a single trunk group, simply use the minus sign (-) in front of the trunk group name and context. Do not use a <Space> between the minus sign and the trunk group name and context.

For example, you might have already configured a list of trunk groups with the following entries: tgrpA: contextA, tgrpB: contextB, tgrpC: contextC, and tgrp1: context1. To delete tgrp1: context1 from the list, you would make the following entry:

```
ACMEPACKET(session-agent)# trunk-group -tgrp1:context1
```

Your list would then contain: tgrpA: contextA, tgrpB: contextB, and tgrpC: contextC.

When you add multiple trunk groups, simple enclose your entry in quotation marks ("") or in parentheses (()). While you put spaces between the trunk group name and context entries, you do not use spaces with the plus sign, parentheses or quotation marks.

```
ACMEPACKET(session-agent)# trunk-group -"tgrp1:context1  
tgrp2:context2"
```

- You overwrite (replace) the entire list of a session agent's trunk groups by entering a list that does not use either the plus (+) or the minus (-) sign syntax.

26. **ping-in-service-response-codes**—*SIP only*. Enter the list of response codes that keep a session agent in service when they appear in its response to the Net-Net SBC's ping request. The Net-Net SBC takes the session agent out of service if a response code is used that does not appear on this list. Default is **none**.
27. **out-service-response-codes**—*SIP only*. Enter the list defines the response codes that take a session agent out of service when they appear in its response to the Net-Net SBC's ping request or any in-dialog creating request (such as an INVITE, SUBSCRIBE, etc.). The Net-Net SBC ignores this list if an in-service list exists.
28. **options**—*Optional*. You can add your own features and/or parameters by using the options parameter. You enter a comma-separated list of either or both of the following:

- feature=<value feature>

For example:

You can include the original address in the SIP message from the Net-Net SBC to the proxy in the Via header parameter by entering the following option:

```
vi a-ori gi n=<parameter-name>
```

The original parameter is included in the Via of the requests sent to the session agent. The via origin feature can take a value that is the parameter name to include in the Via. If the value is not specified for via origin, the parameter name is origin.

Note: If the feature value itself is a comma-separated list, enclose it within quotation marks.

29. **media-profiles**—*Optional and for H.323 only*. You can enter a list of media profiles to open logical channels when starting an outgoing call as a Fast Start H.323 call.

Values you enter here must start with either an alphabetical character from A through Z (AXa-z) or with an underscore (_). After the first character, each list entry can contain any combination of alphabetical or numerical characters (0-9A_Za-z), as well as the period (.), the dash (-), and the underscore (_). For example, netnet_mediaprofile1.

You can enter 1 to 24 characters.

Note: The values you enter here must correspond to a valid name you entered when you configure the media profile.

30. **in-translationid**—*Optional.* Enter the In Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to incoming traffic.
 31. **out-translationid**—*Optional.* Enter the Out Translation ID for a configured session translation (group of address translation rules with a single ID) if you want to apply session translation to outgoing traffic.
- Address translations attached to session agents take precedence over address translations attached to realms. If no address translation is applied to a session agent, then the Net-Net SBC will use the address translation applied to a realm. If an address translation is applied to both a realm and session agent, the translation attached to the session agent will apply. If the applicable session agent and realm have no associated translations, then the addresses will remain in their original forms and no address translations will be performed.
32. **trust-me**—Indicate whether this session agent is a trusted source, which the Net-Net SBC checks when it receives a message to determine if the source is trusted. The default value is **enabled**. The valid values are:
 - enabled | disabled

The following example shows a session agent with an IP address used for the hostname.

session-agent		
hostname	192.168.1.10	
ip-address	192.168.1.10	
port	5060	
state	enabled	
app-protocol	SIP	
app-type		
transport-method	UDP	
realm-id	realm-1	
description	englab	
carriers	carrier1	
allow-next-hop-ip	enabled	
constraints	disabled	
max-sessions	355	
max-inbound-sessions	4	
max-outbound-sessions	355	
max-burst-rate	0	
max-inbound-burst-rate	10	
max-outbound-burst-rate	1	
max-sustain-rate	3000	
max-inbound-sustain-rate	0	
max-outbound-sustain-rate	0	
min-securities	5	
min-asr	0 time-to-resume	60
ttr-no-response	0	
in-service-period	30	
burst-rate-window	60	
sustain-rate-window	3600	
req-uri-carrier-mode	None	
proxy-mode	Proxy	

redirection	Recurse
loose-routing	enabled
send-media-session	enabled
response-map	
ping-method	
ping-interval	0
media-profiles	
in-translational	
out-translational	
trust-me	disabled
request-uri-headers	
stop-recurse	
local-response-map	
ping-to-user-part	
ping-from-user-part	
lli-trust-me	disabled
in-manipulations	
out-manipulations	
p-asserted-id	
trunk-group	
max-register-sustain-rate	0

Configuring Session Agent Groups

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**—Enable or disable the session agent group on the Net-Net SBC. The default value is **enabled**. Valid values are:
 - enabled | disabled
7. **application-protocol**—Indicate the signaling protocol you want to use with the session agent group. The default value is **SIP**. The valid values are:
 - SIP | H.323
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. The default value is **hunt**. The valid values are:
 - **hunt**—Selects session agents in the order in which they are listed. For example, if the first agent is online, working, and has not exceeded defined constraints, all traffic is sent to the first agent. If the first agent is offline or if it exceeds a defined constraint, the second agent is selected. If the first and

second agents are offline or exceed defined constraints, the third agent is selected. And so on through the list of session agents.

- **roundrobin**—Selects each session agent in the order in which they are listed in the destination list, selecting each agent in turn, one per session.
 - **leastbusy**—Selects the session agent that has the fewest number of sessions relative to the maximum outbound sessions constraint or the maximum sessions constraint (for example, lowest percent busy).
 - **propdist**—Based on programmed, constrained session limits, the Proportional Distribution strategy proportionally distributes the traffic among all of the available session agents.
 - **lowsusrate**—The Lowest Sustained Rate strategy routes to the session agent with the lowest sustained rate of session initiations/invitations (based on observed sustained session request rate).
9. **destination**—Identify the destinations (session agents) available for use by this session agent group.
A 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.
 10. **trunk-group**—Enter trunk group names and trunk group contexts to match in either IPTEL or custom format. If left blank, the Net-Net SBC uses the trunk group in the realm for this session agent group. Multiple entries are surrounded in parentheses and separated from each other with spaces.
Entries for this list must one of the following formats: `trgp: context` or `trgp. context`.
 11. **sag-recursion**—Enable this parameter if you want to use SIP SAG recursion for this SAG. The default value is **disabled**. Valid values are:
 - enabled | disabled
 12. **stop-sag-recuse**—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.

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

trunk-group
  tgname2: tgcontext2
  disable
  401, 407
  last-modified-date 2005-01-09 23:23:36

```

SAG Matching for LRT and 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 \(784\)](#) and [CNAM Subtype Support for ENUM Queries \(794\)](#) sections.

Note that you set the state of this feature in the SIP configuration.

To configure a SAG for ENUM or LRT matching:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
 2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#
 3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-config**
ACMEPACKET(sip-config)#
- If you are adding support for this feature to a pre-existing SIP configuration, then you must select (using the ACLI **select** command) that configuration to edit it.
4. **enum-sag-match**—Set this parameter to enabled so the 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.

Configuring Local Policy

To configure local policy:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **local-policy**
ACMEPACKET(local-policy)#
4. **from-address**—Indicate the originating address information by entering a From address value. You can use the asterisk (*) as a wildcard to indicate this policy can be used with all originating addresses.
You can also use complete or partial E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:
 - SIP From address
 - FQDNs

- IP addresses
- H.323 CallingPartyAddress

The Net-Net SBC also supports the asterisk as part of the From address you configure in your local policies.

This means that for the **from-address** parameters of a local policy configuration, you can enter values in which an asterisk appears and match them accordingly. You might enter values that resemble the following examples:

- 123*456
- john*123

5. **to-address**—Indicate the destination address by entering a To address value. You can use the asterisk (*) as a wildcard to indicate all this policy can be used for any destination address.

You can also use E.164 addresses (strings that contain telephone keypad characters) here. Number matching works from left to right. Formats include the following:

- SIP Request-URI
- FQDNs
- IP addresses
- H.323 CalledPartyAddress

The Net-Net SBC also supports the asterisk as part of the To address you configure in your local policies.

This means that for the **to-address** parameters of a local policy configuration, you can enter values in which an asterisk appears and match them accordingly. You might enter values that resemble the following examples:

- 123*456
- john*123

6. **source-realm**—Enter the realm, or list of realms, you want the Net-Net SBC to use to determine how to route traffic. This list identifies from what realm traffic is coming and is used for routing by ingress realm by the local policy.

You can use the asterisk (*) as a wildcard to indicate this local policy can be used with all realms. The default value is *. Or you can enter a value that corresponds to the identifier of an already configured realm. Formats include the following:

- realm ID
- customer name
- peer name
- subdomain name
- VPN identifier

7. **activate-time**—Set the time you want the local policy to be activated using the following syntax:

`yyyy: mm: dd hh: mm: ss`

`yyyy: mm: dd-hh: mm: ss`

8. **deactivate-time**—Set the time you want the local policy to be deactivated using the following syntax:

`yyyy: mm: dd hh: mm: ss`

yyyy: mm: dd-hh: mm: ss

9. **state**—Indicate whether you want the local policy to be enabled or disabled on the system. The default value is **enabled**. The valid values are:

- enabled | disabled

10. **policy-attribute**—Configure local policy attributes by following steps 8 through 21.

11. **next-hop**—Identify the next signaling host by entering the next hop value. You can use the following as next hops:

- IPv4 address of a specific endpoint
- Hostname or IPv4 address of a configured session agent
- Group name of a configured session agent group

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 a request with a 404 Not Found.

12. **realm**—Identify the egress realm (the realm used to reach the next hop) if the Net-Net SBC must send requests out from a specific realm.

The value you enter here must correspond to a valid identifier you enter when you configured the realm. If you do not enter a value here, and the next hop is a session agent, the realm identified in the session agent configuration is used for egress. In H.323, the next hop address is matched against the realm's address prefix to determine the realm.

13. **replace-uri**—Indicate whether you want to replace the Request-URI in outgoing SIP requests with the next hop value.

14. **carrier**—*Optional*. Enter the name of the carrier associated with this route. The value you enter here must match one or more of the carrier names in the session agent configuration.

Entries in carrier fields can be from 1 to 24 characters in length and can consist of any alphabetical character (Aa-Zz), numerical character (0-9), or punctuation mark (! " # \$ % ^ & * () + - = < > ? ' | { } [] @ / \ ' ~ , . _ : ;) or any combination of alphabetical characters, numerical characters, or punctuation marks. For example, both 1-0288 and acme_carri er are valid carrier field formats.

15. **start-time**—Indicate the time of day (from the exact minute specified) the local policy attributes go into effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

1400

The default value of **0000** implies that the defined policy attributes can be considered in effect any time after 00:00:00. The valid range is:

- Minimum—0000
- Maximum—2400

16. **end-time**—Indicate the time of day (from the exact minute specified) the local policy attributes are no longer in effect. Enter only numerical characters (0-9) and follow the 4-digit military time format. For example:

2400

The default value of **2400** implies that the defined policy attributes can be considered in effect any time before midnight. The valid range is:

- Minimum—0000
- Maximum—2400

17. **days-of-week**—Enter any combination of days of the week (plus holidays) you want the local policy attributes to be in effect. You must enter at least one day or holiday here. A holiday entry must correspond with a configured holiday established in the Session Router.

The default is U-S. The valid values are:

- U (Sunday)
- M (Monday)
- T (Tuesday)
- W (Wednesday)
- R (Thursday)
- F (Friday)
- S (Saturday)
- H (Holiday)

You can enter a range of values separated by a hyphen, for example U-S. And you can enter multiple values separated by commas, for example M,W,F. You cannot use spaces as separators.

18. **cost**—Enter a cost value that acts as a unitless representation of the cost of a route relative to other routes reaching the same destination (To address). This value is used as a way of ranking policy attributes.

The default value is zero (0). The valid values are:

- minimum—zero (0)
- maximum—999999999

19. **app-protocol**—Enter the signaling protocol to use when sending messages to the next hop. The valid values are:

- H.323 | SIP

20. **state**—Indicate whether you want to enable or disable the local policy. The default value is **enabled**. The valid values are:

- enabled | disabled

21. **media-profiles**—Configure a list of media profiles if you want the local policy to route SIP and H.323 traffic by the codecs specified in the SDP. The list of media profiles entered here are matched against the SDP included in SIP or H.323 requests and the next hop is selected by codec.

The values in this list are matched against the rtpmap attribute of passed SDP, and preference weight for route selection is based on the order in which the matching payload type appears in the SDP's media (m=) line.

For example when the following SDP arrives:

```
m=audio 1234 RTP/AVP 0 8 18
```

that contains the following attributes that correspond to three configured local policies with the same cost:

- a=rtpmap: 0 PCMU/8000
- a=rtpmap: 8 PCMA/8000
- a=rtpmap: 18 G729/8000

the following route selection action occurs:

The local policy route that corresponds to the `a=rtpmap: 0 PCMU/8000` attribute is selected because the payload type of 0 in the attribute line matches the first payload type of 0 listed in the `m=` line. The codec value of PCMU indicated in this selected attribute is used to find the local policy with the media profiles attribute that includes PCMU in the list.

Because the value you enter here is matched against the codec values included in the actual passed SDP, it must correspond to accepted industry-standard codec values.

The following example shows a local policy with a next hop value of the session agent group called gw-sag2.

local-policy	
from-address	*
to-address	
sourcerealm	192.168.1.10
activatetime	*
deactivatetime	2005-01-20 20:30:00
state	N/A
last-modifieddate	2005-01-10 00:36:29
policy-attribute	
next-hop	gw-sag2
realm	
replace-uri	enabled
carrier	
start-time	0000
end-time	2400
days-of-week	U-S
cost	0
app-protocol	
state	enabled
media-profiles	

Local Policy Matching for Parent Realms

For SIP and H.323, you can configure the 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 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(config)# **session-router**
3. Type **session-router** and press <Enter>.
ACMEPACKET(session-router)# **session-router**
ACMEPACKET(session-router-config)#
4. **match-ip-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
5. Save and activate your configuration.
ACMEPACKET(session-router-config)# **match-ip-src-parent-realms enabled**

SIP Session Agent DNS-SRV Load Balancing

Prior to Release 6.2.0 the Net-Net SBC provided the ability to specify an FQDN (fully qualified domain name) for a destination session-agent. During DNS lookup the FQDN could resolve to multiple SRV (Resource Record for Servers) records. Each SRV could resolve to a single IP address via A-Record query in IMS or DNS.

With Release 6.2.0 the Net-Net SBC supports load balancing behavior as described in RFC 3263, *Session Initiation Protocol (SIP): Locating SIP Servers*.

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 monitor the availability of the dynamically resolved IP addresses obtained from DNS server using OPTIONS ping (ping-per-DNS entry). The *ping-method* and *ping-interval* for each resolved IP addresses will be copied from original session-agent.

Status of Session-Agent:

In Service – if any of dynamically resolved IP addresses is in service

Out of service – if all dynamically resolved IP addresses is out of service.

The default of **ping-all-addresses** is *disabled*, in which case the Net-Net SBC only pings the first available resolved IP addresses.

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-recuse** 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)*.

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

```
2. Use the **ping-all-addresses** parameter to enable Session-Agent DNS-SRV load balancing.
3. Use **done**, **exit**, and **verify-config** to complete Session-Agent DNS-SRV load balancing configuration.

The **show agents** ACLI command displays the availability of dynamically resolved IP addresses

```
ACMEPACKET# show sip agents acme.engr.com
21:46:05-51-router
Session Agent acme.engr.com(core) [In Service] NO ACTIVITY
Session Agent acme.hxu.com(core) [In Service] NO ACTIVITY

Destination: 192.168.200.235 In Service
Destination: 192.168.200.231 In Service
...
...
```

Answer to Seizure Ratio-Based Routing

New SIP and H.323 session agent constraints set a threshold for Answer to Seizure Ratio (ASR) has been implemented. ASR is considered when determining whether session agents are within their constraints to route calls (in addition to session and rate constraints).

The new session agent constraints indicate the minimum acceptable ASR value and computes the ASR while making routing decisions. ASR is calculated by taking the number of successfully answered calls and dividing by the total number of calls attempted (which are known as seizures).

If the ASR constraints are exceeded, the session agent goes out of service for a configurable period of time and all traffic is routed to a secondary route defined in the local policy (next hop with higher cost).

How It Works

The two session agent constraints are:

- minimum seizure: determines if the session agent is within its constraints. When the first call is made to the session agent or the if calls to the session agent are not answered, the minimum seizure value is checked.

For example, if 5 seizures have been made to the session agent and none of them have been answered, the sixth time, the session agent is marked as having exceeded its constraints and the calls will not be routed to it until the time-to-resume has elapsed.

- minimum ASR: considered when make routing decisions. If some or all of the calls to the session agent have been answered, the minimum ASR value is considered to make the routing decisions.

For example, if you set the minimum ASR at 50% and the session agent's ASR for the current window falls below 50%, the session agent is marked as having exceeded its constraints and calls will not be routed to it until the time-to-resume has elapsed.

Configuring ASR Constraints

You can configure ASR constraints using the ACLI or Net-Net EMS.

ACLI Instructions and Examples

To configure ASR constraints:

- In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config terminal**
- Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# **session-router**
- 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 configuring an existing session agent, enter the **select** command to select the session agent.
 - min-seizures**—Enter the minimum number of seizures that when exceeded, cause the session agent to be marked as having exceeded its constraints. Calls will not be routed to the session agent until the time-to-resume has elapsed. The default value is 5. The valid range is:
 - Minimum—1
 - Maximum—999999999

6. **min-asr**—Enter the percentage you want as the minimum. If the session agent's ASR for the current window falls below this percentage, the session agent is marked as having exceeded its constraints and calls will not be routed to it until the time-to-resume has elapsed. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—100
7. Save and activate your configuration.

The following example shows a session agent configuration.

```

session-agent
  hostname          192.168.1.6
  ip-address
  port              1720
  state             enabled
  app-protocol      H323
  app-type          H323-GW
  transport-method
  real-mid
  description
  carriers
  allow-next-hop-ip
  constraints
  max-sessions      0
  max-inbound-sessions   4
  max-outbound-sessions   5
  max-burst-rate     0
  max-inbound-burst-rate 10
  max-outbound-burst-rate 1
  max-sustain-rate   0
  max-inbound-sustain-rate 0
  max-outbound-sustain-rate 0
  min-seizures       5
  min-asr            50
  time-to-resume    30
  ttr-no-response   0
  in-service-period 0
  burst-rate-window 0
  sustain-rate-window 0
  req-uri-carrier-mode None
  proxy-mode
  redirect-action
  loose-routing
  send-media-session
  response-map
  ping-method
  ping-interval     0
  media-profiles
  in-translations
  out-translations
  trust-me          disabled
  request-uri-headers
  stop-recuse
  local-response-map

```

pi ng-to-user-part	
pi ng-from-user-part	
l i -trust-me	di sabl ed
i n-mani pul ati oni d	
out-mani pul ati oni d	
p-asserted-i d	
trunk-group	
max-regi ster-sustai n-rate	0
earl y-medi a-al low	
i nval i date-registrati ons	di sabl ed
l ast-modi fi ed-date	2006-05-12 19: 48: 06

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.

UDP Datagram Support for DNS NAPTR Responses

The Net-Net SBC's default behavior is to conform to the DNS standard defined in RFC 1035 "Domain Names: Implementation and Specification," which sets a maximum size for UDP responses of 512 bytes. This limitation means that responses larger than 512 bytes are truncated (set with the TC, or truncation, bit). In addition, this limitation protects network and system resources because using TCP consumes an undesirable amount of both.

However, you can configure support ENUM queries that manage larger UDP DNS responses as set out in RFC 2671, "Extension Mechanisms for DNS (EDNS0)," enabling your Net-Net SBC to manage responses beyond 512 bytes. According to RFC 2671, senders can advertise their capabilities using a new resource record (OPT pseudo-RR), which contains the UDP payload size the sender can receive. When you specify a maximum response size over 512 bytes, then the Net-Net SBC add the OPT pseudo-RR to the ENUM query—with which the ENUM server will truncate the response.

Custom ENUM Service Type Support

You can configure the ENUM service type that you want to use for an ENUM group. The Net-Net SBC has always supported E2U+si p and si p+E2U by default, and still does. With Release S-C6.1.0, however, you are also able to configure the service type to those supported in RFCs 2916 and 3721.

For example, you can now set the service type in the ENUM configuration to support E2U+si p and E2U+voi cemsg: si p. 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+si p and si p+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 NAPRT, SRV, and A records returned from an ENUM query to eliminate the need for unnecessary external DNS requests by enabling the **cache addl records** parameter. These cached records can then be accessed by internal ENUM and DNS agents.

The unprompted NAPTR, SRV, or A record returned to the 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 perform 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 be configured to include the source URI which caused the ENUM request by enabling the **include source info** parameter. 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.
- Can restrict functionality by specification. This can mean no media management, limited potential for RADIUS accounting, and no CALEA (no Release/BYE messages for CDC).
- Acts primarily as a routing device, with local policy routing and ENUM routing.

Transaction Stateful Proxy

In the transaction stateful proxy mode:

- Adds state to the proxy (not dialogs).
- Has lower number of messages per call. No Record Route header added and no BYES.
- Keeps transaction state (timers and retransmissions).
- Enforces session restrictions (32k) because of state management. These restrictions can be increased.
- Can restrict functionality by specification. This can mean no media management, limited potential for RADIUS accounting, and no CALEA (no Release/BYE message for CDC).
- Acts as routing device with transaction timers, with local policy routing and ENUM routing.
- Can off-load some transactions across unreliable links.

Session Stateful Proxy

The session stateful proxy mode:

- Maintains dialog state as a proxy.
- Includes BYES (though cannot be inserted)
- Keeps transaction state (timers and retransmission)
- Provides per-session information such as session counters per session agent, RADIUS STOP accounting record generation, CALEA CDC generation.
- Enforces session restrictions (32k) because of state management.
- Does not provide media management. There is no CALEA CCC.
- Routes full sessions with transaction timers with local policy routing and ENUM routing.

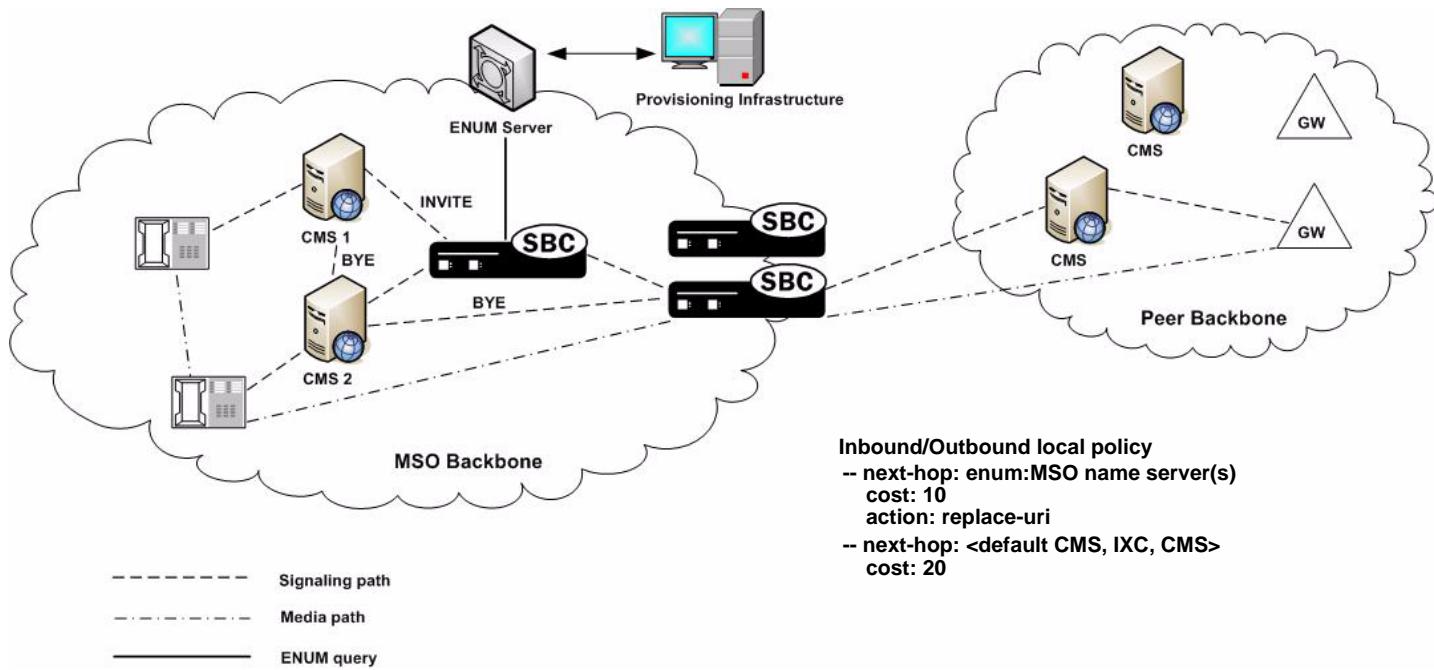
B2BUA

The B2BUA mode:

- Acts as UAS and UAC within call flow.
- Includes BYES (can be inserted).
- Keeps transaction state (timers and retransmissions)
- Provides per-session information such as session counters per session agent, RADIUS STOP accounting record generation, CALEA CDC generation.
- Enforces session restrictions (32k) because of state management.
- Can provide media management, including media routing through a single IP address with topology masking, CALEA CCC, media watchdogs for state management.
- Routes full sessions with topology masking. Includes rewriting Via, Route, Contact headers, full NATing with SIP NAT or header manipulation, direct bridging, local policy routing, and ENUM routing.

Example: ENUM Stateless Proxy

The following diagram shows the 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

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 4.4.4.4.3.3.1.8.7.1.e164.arpa.com.

6. **realm-id**—Enter the realm where the ENUM servers can be reached. The realm ID is used to determine on which network interface to issue the ENUM query.

7. **enum-servers**—Enter the list of ENUM servers (an ENUM server and corresponding redundant servers) to be queried. Separate each server address with a space and enclose list within parentheses.

The first server on this list is the first one to be queried. If the query times out (including retransmissions) without getting a response, the next server on the list is queried and so on.

8. **service-type**—Enter the ENUM service types you want supported in this ENUM configuration. Possible entries are E2U+si p and si p+E2U (the default), and the types outlined in RFCs 2916 and 3721.

This parameter defaults to the following service types: E2U+si p and si p+E2U.

You can enter multiple services types in the same entry, as in this example:

```
ACMEPACKET(enum-config)# service-type E2U+si p, si p+E2U, E2U+vol cemsg
```

9. **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.

10. **timeout**—Enter the total time in seconds that should elapse before a query sent to a server (and its retransmissions) will timeout. If the first query times out, the next server is queried and the same timeout is applied. This process continues until all the servers in the list have timed out or until one of the servers responds.

The retransmission of ENUM queries is controlled by three timers. These timers are derived from this timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the Net-Net SBC should retransmit 3 times before timing out to give the server a chance to respond. The valid values are:

- **Init-timer**—Is the initial retransmission interval. If a response to a query is not received within this interval, the query is retransmitted. To safeguard from performance degradation, the minimum value allowed for this timer is 250 milliseconds.
- **Max-timer**—Is the maximum retransmission interval. The interval is doubled after every retransmission. If the resulting retransmission interval is greater than the value of max-timer, it is set to the max-timer value.
- **Expire-timer**—Is the query expiration timer. If a response is not received for a query and its retransmissions within this interval, the server will be considered non-responsive and the next server in the list will be tried.

The following examples show different timeout values and the corresponding timers derived from them.

timeout >= 3 seconds

```
Init-timer = Timeout/11
Max-Timer = 4 * Init-timer
Expire-Timer = Timeout
```

timeout = 1 second

```

Init-Timer = 250 ms
Max-Timer = 250 ms
Expire-Timer = 1 sec
timeout = 2 seconds
Init-Timer = 250 ms
Max-Timer = 650 ms
Expire-Timer = 2sec

```

11. **cache-inactivity-timer**—Enter the time interval in seconds after which you want cache entries created by ENUM requests deleted, if inactive for this interval. If the cache entry gets a hit, the timer restarts and the algorithm is continued until the cache entry reaches its actual time to live. Setting this value to zero disables caching. For optimal performance, set this to one hour. Rarely used cache entries are purged and frequently used entries are retained. The default value is **3600**. The valid range is:
 - Minimum—0
 - Maximum—999999999
12. **lookup-length**—Specify the length of the ENUM query, starting from the most significant digit. The default is **0**. The valid range is:
 - Minimum—1
 - Maximum—255
13. **max-response-size**—Enter the maximum size in bytes for UDP datagrams in DNS NAPTR responses. This parameter takes values from 512 (default) to 65535. Although the maximum value you can set is 65535, Acme Packet recommends configuring values that do not exceed 4096 bytes. For more information about response UDP datagram response size, refer to the [UDP Datagram Support for DNS NAPTR Responses \(785\)](#) section above.
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. Save your work.

Example

The following example shows an ENUM configuration called enumconfig.

enum-config	
name	enumconfig
top-level-domain	
realm-id	public
enum-servers	10.10.10.10:3456 10.10.10.11
service-type	E2U+sip, si p+E2U
query-method	hunt

timeout	11
cachelactivitytimer	3600
max-response-size	512
health-query-number	+17813245678
health-query-interval	0
failover-to	enumconfig2
cache-addl-records	enabled
include-source-info	disabled

Configuring the Local Policy Attribute

You can specify that an ENUM query needs to be done for the routing of SIP calls. You do so by configuring the local policy's next-hop attribute with the name of a specific ENUM configuration, prefixed with the enum: tag. For example: enum: test

You can configure multiple next-hops with different ENUM servers or server groups (possibly with different top-level-domains). If the first ENUM server group you enter as the next hop is not available, one of the others can be used.

Note: A new parameter called **action** has replaced the policy attribute's **replace-uri** parameter available prior to build 211p19.

To configure local policy:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
3. Type **local-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **local-policy**
ACMEPACKET(local-policy)#
4. **next-hop**—Enter the name of the ENUM configuration with the prefix enum:. For example, enum: test.
5. **action**—Set to **redirect** if you want to send a REDIRECT message back to the calling party with the information returned by ENUM in the Contact. The calling party then needs to send a REDIRECT using that information. The default value is **none**. Valid values are:
 - **none**—No specific actions requested.
 - **replace-uri**—To replace the next Request-URI with the next hop.
 - **redirect**—To send a redirect response with this next hop as contact.
6. Save and activate your configuration.

Local Policy Example

The following example shows one local policy with the next-hop configured to use enum:test and a second with the next-hop configured to use enum:test_alternate.

local-policy

from-address	*
to-address	*
source-realname	public
activate-time	N/A
deactivate-time	N/A

state	enabled
last-modified-date	2006-03-09 09:18:43
policy-attribute	
next-hop	enum: test
realm	public
action	none
terminate-recursion	disabled
carrier	
start-time	0000
end-time	2400
days-of-week	U-S
cost	1
app-protocol	SIP
state	enabled
media-profiles	
policy-attribute	
next-hop	enum: test_alternate
realm	public
action	none
terminate-recursion	disabled
carrier	
start-time	0000
end-time	2400
days-of-week	U-S
cost	2
app-protocol	SIP
state	enabled

CNAM Subtype Support for ENUM Queries

CNAM, calling name, data is a string up to 15 ASCII characters of information associated with a specific calling party name. The Internet-draft, draft-ietf-enum-cnam-08.txt, registers the Enumservice 'pstndata' and subtype 'cnam' using the URI scheme 'pstndata:' to specify the return of CNAM data in ENUM responses. The 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 perform 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 perform 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 Configuration and Examples

To enable the Net-Net SBC to perform 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# config terminal
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.
ACMEPACKET(config)# 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 \(804\)](#) 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.

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.

ACLI Instructions and Examples

Setting Up a Local Routing Configuration

This section shows you how to:

- Set up local route configuration
- Specify that a set of local policy attributes needs to use local routing

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/routing directory. There is no default for this parameter and it is required.
6. **prefix-length**—Enter the number of significant digits/bits to be used for lookup and cache storage. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—999999999
7. Save and activate your configuration.

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.

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:

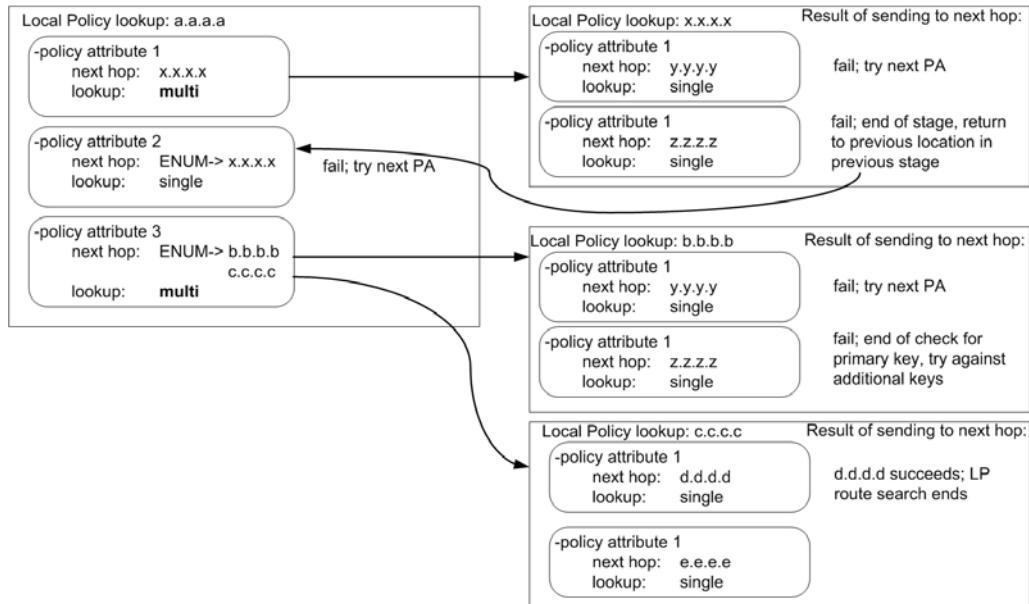
- An operator might need to query an ENUM server for a destination number. Based on the NAPTR result of the ENUM query, the Net-Net SBC performs a local policy lookup to decide how to route the request, perhaps based on a LRT table lookup.
- An operator might need to query one ENUM server for a number 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	61768000000
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

The local route table in default-lrt appears as follows:

```
<route>
<user type="E164">159</user>
<next type="regex">! ^ .*$! si p: 11568000000@192. 168. 200. 47! </next>
<next type="regex">! ^ .*$! si p: 21568000000@192. 168. 200. 99! </next>
<next type="regex">! ^ .*$! si p: 11578000000@192. 168. 200. 44! </next>
</route>
```

1. The Net-Net SBC receives an INVITE on realm, private (SDP is omitted below):

```
I INVITE si p: 159@192. 168. 1. 49: 5060 SIP/2. 0
Via: SIP/2. 0/UDP 192. 168. 1. 48: 5060
From: si pp <si p: si pp@192. 168. 1. 48: 5060>; tag=1
To: sut <si p: 159@192. 168. 1. 49: 5060>
Call-ID: 1-4576@192. 168. 1. 48
CSeq: 1 INVITE
Contact: si p: si pp@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: sip <sip:sip@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:

Irt: default-Irt
 192.168.200.50
 Irt: emergency
 Irt: carrier-Irt

- 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-Irt 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 =sip <sip:sip@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 =sip <sip:sip@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:

```
I INVITE si p: 215680000002@192. 168. 200. 98; lr SIP/2. 0
Via: SIP/2. 0/UDP 192. 168. 200. 49: 5060
From: si pp <si p: si pp@192. 168. 1. 48: 5060>
To: sut <si p: 159@192. 168. 1. 49: 5060>
Call-ID: SDnhae701-76e8c8b6e168958e385365657faab5cb-v3000i 1
CSeq: 1 INVITE
Contact: <si p: si pp@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 is for this parameter is the same as with the Routing-based RN and CIC feature.

Multistage Routing Lookup Termination

It is important for the 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# config terminal
ACMEPACKET(config)#

```
2. Type **session-router** and press <Enter>.

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

```
3. Type **local-policy** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit a local policy.

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

```
4. Type **policy-attributes** and press <Enter>. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

```
ACMEPACKET(local-policy)# policy-attributes
ACMEPACKET(local-policy-attributes)#

```
5. **next-hop**—This is the next signaling host and/or object to query. This parameter can be configured as an IP address, ENUM server, or LRT. You can also add a lookup key to an ENUM server or LRT lookup with the following syntax:

```
next-hop enum:ENUM-object;key=$TO

```
6. **terminate-recursion**—Set this parameter to **enabled** to terminate local policy route recursion when the current stage completes.
7. **lookup**—Leave this parameter at the default **single** for single stage local policy routing or set it to **multi** to enable multistage local policy routing.

8. **next-key**—Set this parameter to \$TO, \$FROM, or \$PAI if you wish to override the recently-returned lookup key value for the next stage.
9. Save and activate your configuration.

Maintenance and Troubleshooting

The **show sipd policy** command includes four additional counters that refer to single and multistage local policy lookups. All counters are reported for the recent period, and lifetime total and lifetime period maximum. These counters are:

- Local Policy Inits—Number of times the 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.

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 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

Setting the Lookup Key

This section shows you how to specify that a set of local policy attributes should use an RN for lookup. You can also set this value to CIC, or to any value you require.

You can set the lookup key to an RN in the local policy attributes' **next-hop** parameter.

To set the lookup key to RN:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(config)# **session-router**
3. Type **local-policy** and press <Enter>.
ACMEPACKET(session-router)# **local-policy**
ACMEPACKET(local-policy)#
4. Type **policy-attributes** and press <Enter>.
ACMEPACKET(local-policy)# **policy-attributes**
ACMEPACKET(local-policy-attributes)#
5. **next-hop**—In the **next-hop** parameter—after the kind of ENUM service used—type a colon (:). Then, without spaces, type in **key=rn** and press <Enter>.
ACMEPACKET(local-policy-attributes)# **next-hop lrt:lookup; key=rn**
6. Save and activate your configuration.

Codec Policies for SIP

The Net-Net SBC now has the ability to strip and reorder codecs for SIP sessions. This builds on the Net-Net SBC's pre-existing abilities to route by codec and re-order one codec in an SDP offer by allowing you to configure the order of multiple codecs and to remove specific codecs within the media descriptions in SDP offers.

You can enable the Net-Net SBC to perform these operations on SDP offers by configuring codec policies. Codec policies are sets of rules that specify the

manipulations to be performed on SDP offers and answers. They are applied on an ingress and egress basis using the realm and session agent configurations.

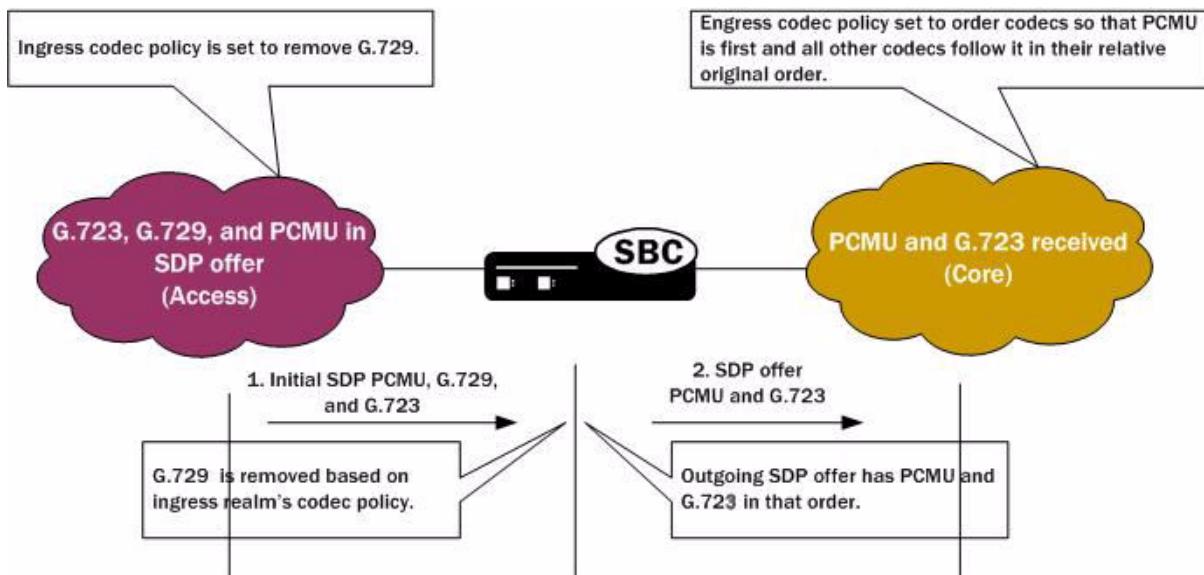
How It Works

There are two types of codec policies that the Net-Net SBC applies:

- Ingress policy—Codec policy that the Net-Net SBC applies to the SDP offer for incoming traffic
- Egress policy—Codec policy that the Net-Net SBC applies to the SDP offer for traffic leaving the Net-Net SBC

The Net-Net SBC applies codec policies during the offer phase of media format negotiation. If codec manipulation is enabled, then the Net-Net SBC performs the modification according to the specific policy and forwards on the traffic.

For example, when the Net-Net SBC receives a SIP INVITE with SDP, it refers to the realm through which the INVITE arrived and performs any codec manipulations (specified in the ingress policy) that have been applied to that realm. With the media description changed according to the ingress policy, the Net-Net SBC passes it to the outgoing realm so that the egress policy can be applied. When the second, egress policy has been applied, the Net-Net SBC forwards on the INVITE.



Since the offer-answer exchange can occur at different stages of SIP messaging, the assigned ingress and egress roles follow the media direction rather than the signaling direction. It might be, for example, that the offer is in an OK that the Net-Net SBC modifies.

You can apply codec policies to realms and to session agents; codec policies configured in session agents take precedence over those applied to realms. However, it is not required that there be both an ingress and an egress policy either for realms or for session agents. If either one is unspecified, then no modifications take place on that side. If there are neither ingress nor egress policies specified, then this feature is disabled and the Net-Net SBC will behave as it prior to this feature's introduction.

Relationship to Media Profiles

For each codec that you specify in a codec policy, there must be a corresponding media profile configuration on the Net-Net SBC. You configure media profiles in the ACLI via the session-router path. In them, you can specify codec type, transport protocol, required bandwidth, and a number of constraints.

Manipulation Modes

You can configure a codec policy to perform several different kinds of manipulations:

- Allow—List of codecs that are allowed for a certain codec policy; if a codec does not appear on this list, then the Net-Net SBC removes it. You can wildcard this list with an asterisk (*) so that all codecs are allowed. Further, you can create exceptions to a wildcarded allow list.

- You make an exception to the wildcarded list of codecs by entering the codec(s) that are not allowed with a **no** attribute. This tells the Net-Net SBC to allow all codecs except the one(s) you specify.

```
ACMEPACKET(codec-policy)# allow-codecs * PCMA: no
```

- You can also create exceptions to allow lists such that audio or video codecs are removed. However, when the allow list specifies the removal of all audio codecs and an INVITE arrives at the Net-Net SBC with only audio codecs, the Net-Net SBC behaves in accordance with RFC 3264. This means that the resulting SDP will contain one attribute line, with the media port for the media line set to 0. The terminating side will need to supply new SDP in its reply because the result of the manipulation is the same as an INVITE with no body.

```
ACMEPACKET(codec-policy)# allow-codecs * audio: no
```

- Order—List of the codecs where you specify their preferred order in the outgoing media offer. The Net-Net SBC arranges matching codecs according to the rule you set, and any remaining ones are added to the list in the same relative order they took in the incoming media offer. If your list specifies a codec that is not present, then the ordering proceeds as specified but skips the missing codec.

You can use an asterisk (*) as a wildcard in this list, too. The placement of the asterisk is key, as you can see in the following examples:

- For an order rule set this way

```
ACMEPACKET(codec-policy)# order A B C *
```

codecs A, B, and C will be placed at the front of the codec list in the order specified; all other codecs in the offer will follow A, B, and C in the same relative order they had in the original SDP offer.

- For an order rule set this way:

```
ACMEPACKET(codec-policy)# order * A B C
```

codecs A, B, and C will be placed at the end of the codec list in the order specified; all other codecs in the offer will come before A, B, and C in the same relative order they had in the original SDP offer.

- For an order rule set this way

```
ACMEPACKET(codec-policy)# order A * B C
```

codec A will be placed at the beginning of the codec list, to be followed by all other codecs in the offer in the same relative order they had in the original SDP offer, and then B and C will end the list.

- Force—An attribute you can use in the allow list with one codec to specify that all other codecs should be stripped from the outgoing offer. You can specify multiple forced codecs in your rules.

- If you set multiple codecs in the allow list and one of them is forced, then the outgoing offer will contain the forced codec.
- If you set multiple codecs in the allow list and the one that is forced is not present in the offer, then the Net-Net SBC will select a non-forced codec for the outgoing offer.

```
ACMEPACKET(codec-policy)# allow PCMU G729:force
```

You cannot use the force attribute with a wildcarded allow list.

- **No**—An attribute that allows you to strip specified codecs or codec types from a wildcarded allow list.

```
ACMEPACKET(codec-policy)# allow * PCMA: no
```

In-Realm Codec Manipulation

In addition to being able to apply codec policies in realms, the realm configuration supports a setting for determining whether codec manipulation should be applied to sessions between endpoints in the same realm.

In-realm codec manipulation can be used for simple call flows that traverse two realms. If the originating and terminating realms are the same, the Net-Net SBC checks to see if you have enabled this capability. If you have enabled it, then the Net-Net SBC performs the specified manipulations. If this capability is not enabled, or if the realm's media management in realm (**mm-in-realm**) setting is disabled, then the Net-Net SBC does not perform codec manipulations.

For more complex calls scenarios that involve call agent or reinitiation of a call back to the same realm, the Net-Net SBC does not perform in-realm codec manipulation.

ACLI Instructions and Examples

Creating a Codec Policy

This section gives instructions and examples for how to configure codec policies and then apply them to realms and session agents. It also shows you how to configure settings for in-realm codec manipulation.

To create a codec policy:

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

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

```
ACMEPACKET(config)# media-manager
```
3. Type **codec-policy** and then press <Enter>.

```
ACMEPACKET(media-manager)# codec-policy
```
4. **name**—Enter the unique name for the codec policy. This is the value you will use to refer to this codec policy when you apply it to realms or session agents. This parameter is required and is empty by default.

5. **allow-codecs**—Enter the list of media format types (codecs) to allow for this codec policy. In your entries, you can use the asterisk (*) as a wildcard, the force attribute, or the no attribute so that the allow list you enter directly reflect your configuration needs. For more information, refer to the [Manipulation Modes \(807\)](#) section above.

The codecs that you enter here must have corresponding media profile configurations.

6. **order-codecs**—Enter the order in which you want codecs to appear in the outgoing SDP offer. Remember that you can use the asterisk (*) as a wildcard in different positions of the order to directly reflect your configuration needs. For more information, refer to the [Manipulation Modes \(807\)](#) section above.

The codecs that you enter here must have corresponding media profile configurations.

7. Save and activate your configuration.

Your codec policy configuration will resemble the following example:

```
codec-policy
  name          private
  allow-codecs  g723: no pcmu video: no
  order-codecs  pcmu
```

Applying a Codec Policy to a Realm

Note that codec policies defined for session agents always take precedence over those defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **media-manager**
3. Type **realm-config** and press <Enter>.
ACMEPACKET(media-manager)# **realm-config**
If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **codec-policy**—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

Applying a Codec Policy to a Session Agent

Note that codec policies that are defined for session agents always take precedence over those that are defined for realms.

To apply a codec policy to a realm:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**

3. Type **session-agent** and press <Enter>.
ACMEPACKET(session-router)# session-agent
 If you are adding support for this feature to a pre-existing session agent, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **codec-policy**—Enter the name of the codec policy that you want to apply to this realm. By default, this parameter is empty.
5. Save and activate your configuration.

In-Realm Codec Manipulations

To enable in-realm codec manipulations:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure terminal
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# media-manager
3. Type **realm-config** and press <Enter>.
ACMEPACKET(media-manager)# realm-config
 If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI **select** command) the realm that you want to edit.
4. **codec-manip-in-realm**—Enter the name of the codec policy that you want to apply to this realm. The default value is **disabled**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

QoS Based Routing

In addition to configuring your system for routing based on certain session constraints, you can also set up routing based on QoS. QoS based routing uses the R-Factor on a per-realm basis to either cut back on the traffic allowed by a specific realm, or to shut that traffic off altogether.

To use this feature, you set up QoS constraints configurations and apply one per realm. The QoS constraints configuration allows you to set up two thresholds:

- Major—The major threshold sets the R-Factor limit beyond which the Net-Net SBC rejects a certain percentage (that you configure) of calls. That is to say, it rejects inbound calls at the rate you set with a **503 Service Unavailable** status code, and rejects outbound calls if there are no alternative routes.
- Critical—The critical threshold, when exceeded, causes the Net-Net SBC to behave the same way it does when any of the session constraints (set in the session-constraints configuration) are exceeded. All inbound calls to the realm are rejected with a **503 Service Unavailable** status code, and (if there is no alternate route) outbound calls are rejected, too. Until the R-Factor falls within acceptable means and the session constraint's time-to-resume value has elapsed, the realm remains in this state.

Management

This feature is supported by MIBs and traps; for more information, refer to the *Net-Net 4000 MIB Reference Guide*. Historical data recording (HDR) also supports this feature by providing the following metrics in the session realm statistics collection group:

- Average QoS RFactor (0-93)
- Maximum QoS RFactor (0-93)
- Current QoS Major Exceeded
- Total QoS Major Exceeded
- Current QoS Critical Exceeded
- Total QoS Critical Exceeded

ACLI Instructions and Examples

Configuring QoS Constraints

Your first step to enabling QoS based routing is to set up a QoS constraints configuration. This configuration is where you enter major and critical thresholds, as well as the load reduction for the realm should the R-Factor exceed the major threshold.

To set up a QoS constraints configuration:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **qos-constraints** and press <Enter>.

```
ACMEPACKET(session-router)# qos-constraints
ACMEPACKET(qos-constraints)#
```
4. **name**—Enter the name of this QoS constraints configuration. This parameter uniquely identifies the configuration, and you use this value when applying the configuration to a realm. This parameter has no default and is required.
5. **state**—Set the state of this QoS constraints configuration. The default is **enabled**, but you can set this parameter to **disabled** if you want to stop applying these constraints.
6. **major-rfactor**—Enter a numeric value between **0** (default) and **9321** to set the threshold that determines when the Net-Net SBC applies the call reduction rate. If you leave this parameter set to **0**, then the Net-Net SBC will not apply a major threshold for any realm where you apply this QoS constraints configuration.

Note that this value must be less than that you set for the **critical-rfactor**, except when the **major-rfactor** is **0**.
7. **critical-rfactor**—Enter a numeric value between **0** (default) and **9321** to set the threshold that determines when the Net-Net SBC rejects all inbound calls for the realm, and rejects outbound calls when there is no alternate route. If you leave this parameter set to **0**, then the Net-Net SBC will not apply a critical threshold for any realm where you apply this QoS constraints configuration.

Note that this value must be greater than that you set for the **major-rfactor**, except when the **major-rfactor** is 0.

8. **call-load-reduction**—Enter a number from 0 (default) to 100 representing the percentage by which the Net-Net SBC will reduce calls to the realm if the **major-rfactor** is exceeded. If you leave this parameter set to 0, then the Net-Net SBC will not reduce call load for the realm—even when the major-rfactor is configured.

This is the percentage of inbound and outbound calls the Net-Net SBC will reject. For example, if you set this parameter to 50 and the major threshold is exceeded, then the Net-Net SBC rejects every other call to the realm.

9. Save and activate your configuration.

Applying QoS Constraint to a Realm

You apply QoS constraints to realms using the **qos-constraint** parameter.

To apply a QoS constraint to a realm:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>

```
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **realm-config** and press <Enter>. If you adding this feature to a pre-existing realm, then you need to select and edit that realm.

```
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(real-m-config)#
```
4. **qos-constraints**—Enter the name value from the QoS constraints configuration you want to apply to this realm.

Save and activate your configuration.

Introduction

This chapter describes the Net-Net SBC's number translations feature.

About Number Translation

Net-Net SBC number translation is used to change a layer-5 endpoint name according to prescribed rules. Number translations can be performed on both the inbound and the outbound call legs independently, before and after routing occurs. Number translation is used for SIP, H.323, and SIP/H.323 interworking configurations.

Number translation takes place twice for both H.323 and SIP calls. The first number translation is applied to the incoming leg of the call, before the outgoing route is selected. The second number translation is applied to the outgoing leg of the call after the outgoing route is selected.

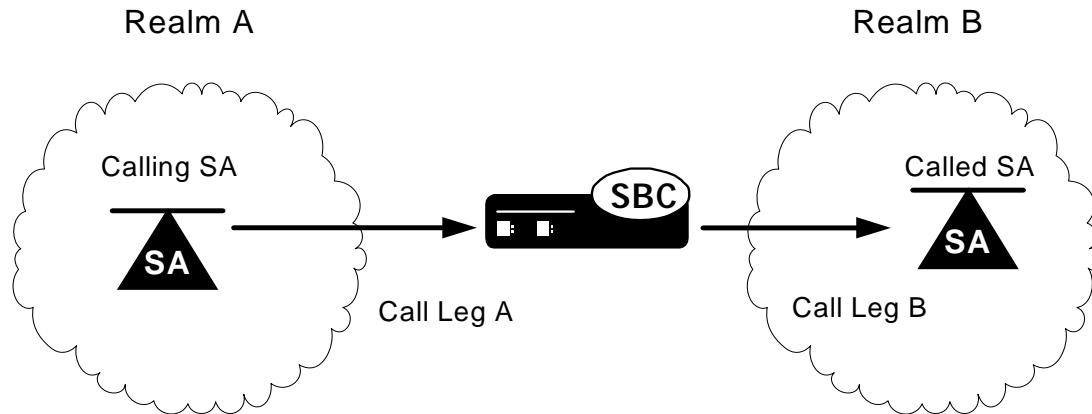
Number translation can be used to strip address prefixes added by external gateways. It can also be used to add a string tag to an address in order to implement a local policy routing scheme, and then remove the tag upon egress from the Net-Net SBC. The most common use of number translation is to add or remove a "1" or a "+" from a phone number sent from or addressed to a device.

Number Translation Implementation

Net-Net SBC 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 SBC 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 SBC.

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 SBC. The following diagram illustrates the number translation concept.



The following table shows you which parameters to apply a session translation ID in order to affect the corresponding leg of the call as shown in the illustration.

Leg	Calling SA	Called SA	Realm A	Realm B
A	IN Translation ID		IN Translation ID	
B		OUT Translation ID		OUT Translation ID

Number Translation in SIP URIs

Number translations only change the user portion of the URI. A typical SIP URI looks like `sip:user@hostname`. The user portion can take the form of either a phone number or any other string used for identification purposes.

Within the SIP header exists a Request URI, a To URI, and a From URI. The session translation element's rules calling parameter modifies the From URI, while the rules called parameter modifies the Request URI and the To URI.

Session Translation in H.323 Messages

Because H.323 messages explicitly define the calling and called parties, the correspondence is exactly the same between the endpoints and configuration parameters. The H.323 calling party corresponds to the session translation element's rules calling parameter. The H.323 called party corresponds to the session translation element's rules called parameter.

Number Translation Configuration Overview

This section describes the procedure to create and apply number translations on the Net-Net SBC.

Configuring the number translation feature requires the following steps:

1. Configure individual translation rules in the translation rules element.
2. Group these rules for use in the session translation element.
3. Apply these groups of rules on a per session agent or per realm basis using the appropriate fields in the session agent or realm configuration elements.

Translation Rules

The translation rules subelement is where the actual translation rules are created. The fields within this element specify the type of translation to be performed, the addition or deletion to be made, and where in the address that change takes place. Translations are not applied to any realm or session agent in this element.

When creating translation rules, first determine the type of translation to perform. The following table lists and describes the three types of number translations.

Field Value	Description
add	This translation type adds a character or string of characters to the address.
delete	This translation type deletes a character or string of characters from the address.
replace	This translation type replaces a character or string of characters within the address. Replace works by first applying the delete parameter then by applying the add parameter.

After you set the translation type, you define the string to add or delete. The wildcard term for a string to delete is the at-sign, "@". Finally, you specify the character position in the address to make the addition or deletion.

The character position where an add or delete occurs is called an index. The index starts at 0 (immediately before the leftmost character) and increases by 1 for every position to the right you move. In order to specify the final position in an address, use the dollar-sign, "\$".

To create a translation rule that deletes a string:

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 a 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 SBC 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 SBC. 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	
i d	addpl us1
type	add
add-string	+1
add-index	0
delete-string	
delete-index	0

Translation Rules

Set the following parameters to configure a translation rule:

1. **ID**—Set a descriptive ID name for this translation rule.
2. **type**—Set the type of translation rule you want to configure. The default value is **none**. The valid values are:
 - **add**—Adds a character or string of characters to the address
 - **delete**—Deletes a character or string of characters from the address
 - **replace**—Replaces a character or string of characters within the address
 - **none**—Translation rule is disabled
3. **add-string**—Enter the string to be added during address translation to the original address. The value in this field should always be a real value; i.e., this field should not be populated with at-signs (@) or dollar-signs (\$). The default value is a blank string.
4. **add-index**—Enter the position, 0 being the left most position, where you want to add the string defined in the **add-string** parameter. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999

5. **delete-string**—Enter the string to be deleted from the original address during address translation. Unspecified characters are denoted by the at-sign symbol (@).

When the type is set to **replace**, this value is used in conjunction with the add-string value. The value specified in the delete-string field is deleted and the value specified in the add-string field is inserted. If no value is specified in the delete-string parameter and the type field is set to replace, then nothing will be inserted into the address. The default value is a blank string.

6. **delete-index**—Enter the position, 0 being the left most spot, where you want to delete the string defined in the **delete-string** parameter. This parameter is only used if the **delete-string** parameter is set to one or more at-signs. The default value is zero (0). The valid range is:

- Minimum—0
- Maximum—999999999

Session Translation

To configure session translations:

1. Exit out of the translation rules element and enter the session translation element.

```
ACMEPACKET(translation-rules)# exit
ACMEPACKET(session-router)# session-translation#
ACMEPACKET(session-translation)#+
```

From this point, you can configure the session translation element. To view all session translation parameters, enter a ? at the system prompt. The following is an example what a session translation configuration might look like:

session-translation	id	rules-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

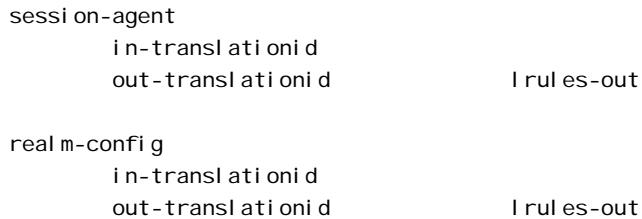
To move from the session-translation element to the realm-config element:

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.



**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 SBC'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 SBC maps FQDNs that appear in certain headers of incoming SIP messages to the IPv4 address that the Net-Net SBC 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.

## Overview

---

This chapter describes how to configure the Net-Net SBC 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 SBC 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 SBC is not directly connected, you need to define multiple realms per network interface.

### Bandwidth-Based Admission Control

The Net-Net SBC is a policy enforcement point for bandwidth-based call admission control. Sessions are admitted or rejected based on bandwidth policies, configured on the Net-Net SBC 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 SBC 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 SBC determines the required bandwidth from the SDP/H.245 information for SIP and from the OLC sent in the SETUP message for H.323. Any request that would cause the bandwidth constraint to be exceeded is rejected with a SIP 503 Service Unavailable or an H.323 Release Complete.

For example, if an incoming SIP message requests PCMU for a payload/encoding name, a zero (0) payload type, and an 8000 cycle clock rate, the Net-Net SBC 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 SBC rejects the session.

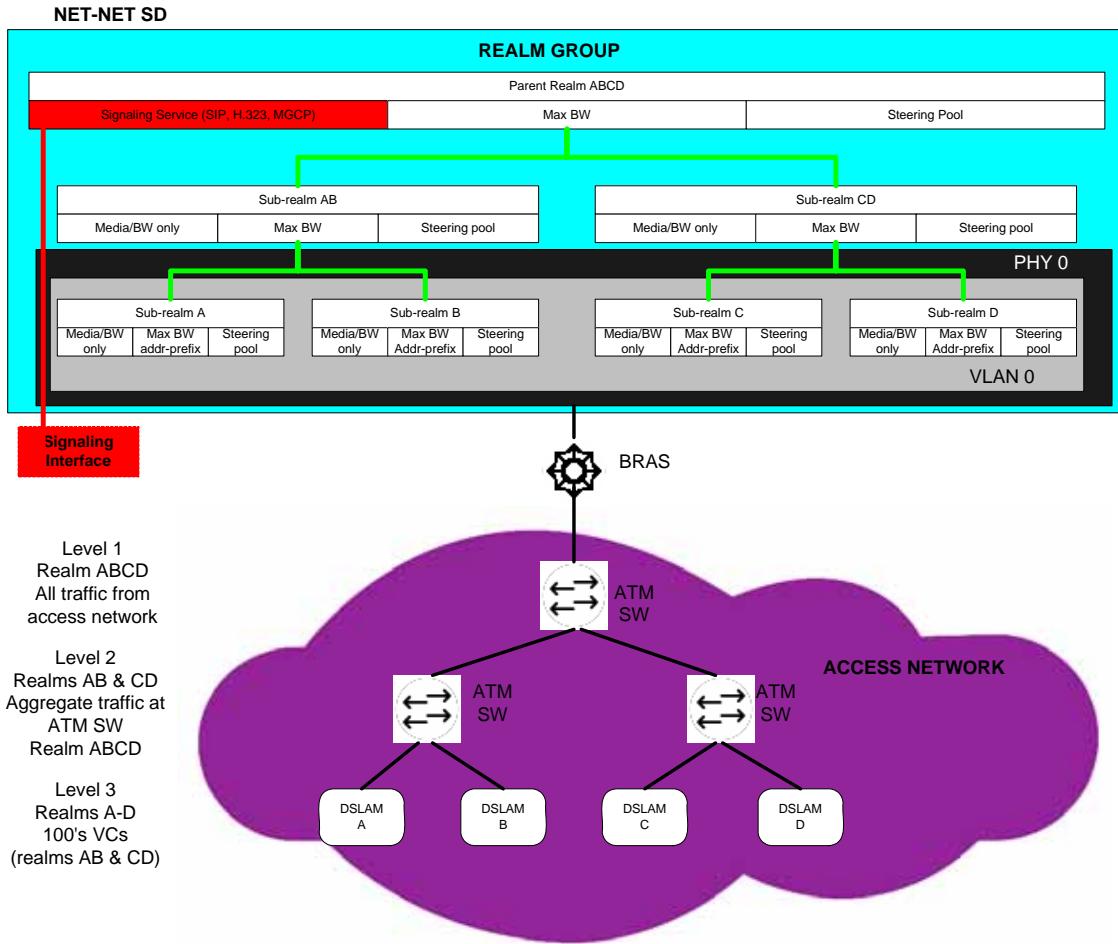
With these mechanisms, the Net-Net SBC provides bandwidth-based admission control.

### **Multi-Level Bandwidth Policy Nesting**

Multi-level nesting of bandwidth policy enforcement addresses the following issues:

- Bandwidth over-subscription: access or transit transport networks are aggregated and/or oversubscribed. For example, digital subscriber lines (DSL), Frame Relay (FR), and Asynchronous Transfer Mode (ATM). Admission control policies must reflect access network topology.
- Bandwidth partitioning for multiple services: access or transit bandwidth is partitioned among multiple service profiles (for example, SIP and MGCP) in the same customer network.
- Multi-site VPN environments: admission control must be applied at the site level as well as the VPN level.

The following example illustrates different scenarios; in each there are two or more levels of admission control required. Nested admission control is best depicted by the DSL broadband example.



In DSL access networks, ATM network bandwidth is typically oversubscribed at rates up to 400/1. At Level 3 (above), hundreds of users virtual circuits (VCs) are aggregated to a smaller set of virtual paths (VPs) at each DSLAM. At Level 2, many virtual paths are aggregated at the first ATM switch. Finally, at Level 1, all traffic from all subscribers in the access network is aggregated at the BRAS. Each level of aggregation is oversubscribed, creating the need to perform admission control at each level.

From a Net-Net SBC 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 SBC to determine in which realm a user resides. When a call request to or from a particular user is received, the Net-Net SBC 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 SBC only accepts requests from configured session agents. And you can set up session admission control so that the Net-Net SBC limits the number of concurrent inbound and outbound sessions for any known service element.

The Net-Net SBC denies a call request to any destination that has exceeded its configured policies for session capacity and session rate. The Net-Net SBC might reject the call request back to the originator. If multiple destinations are available, the Net-Net SBC will check current capacity and rate for each destination and attempt to route the call only to destinations whose policy limits have not been reached.

You assign a media profile to a session agent and indicate whether the transport protocol is SIP or H.323. If the protocol is H.323, you need to indicate whether the session agent is a gateway or a gatekeeper.

## Constraints for Proxy Mode

The Net-Net SBC applies session router and session agent constraints when it is in proxy (transaction or stateless) mode if you enable the ACLI **constraints** parameter for a session agent. However, the Net-Net SBC does not track SIP sessions when in transaction mode, so the following session-specific constraints are not applied:

- max-sessions
- max-inbound-sessions
- max-outbound-sessions
- min-seizures
- min-asr

Constraints the Net-Net SBC applies are:

- max-burst-rate
- max-inbound-burst-rate
- max-outbound-burst-rate
- max-sustain-rate
- max-inbound-sustain-rate
- max-outbound-sustain-rate

In order to set the desired time windows for computing burst rates and sustain rates, you also need to configure these parameters in the session agent configuration: **burst-rate-window** and **sustain-rate-window**. You can also set the time-to-resume and in-service-period parameters to control how long to wait before bringing a session agent back into service after its constraints are no longer exceeded.

## CAC, Policing, and Marking for non-Audio/non-Video Media

The 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.

## **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# config terminal
ACMEPACKET(config)#
```
2. Type `media-manager` and press <Enter>.  

```
ACMEPACKET(config)# 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(real-m-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# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.  

```
ACMEPACKET(config)# 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(medi a-manager)# real m-confi g
ACMEPACKET(real m-confi g)#
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.
```

Admission Control for Session Agents

This section explains how to configure 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**—Enable this parameter. From here you can either configure admission control based on session capacity, session rates, or both. The default value is **enabled**. The valid values are:
 - enabled | disabled

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 default value is zero (**0**). The valid range is:
 - Minimum—0
 - Maximum— $2^{32} - 1$
3. **max-inbound-sessions**—Enter the maximum number of inbound sessions allowed from this session agent. The default value is zero (**0**). The valid range is:
 - Minimum—0

- Maximum—999999999
4. **max-outbound-sessions**—Enter the maximum number of concurrent outbound sessions (outbound from the Net-Net SBC) that are allowed from this session agent. The default value is zero (0). The valid range is:
- Minimum—0
 - Maximum— $2^{32} - 1$

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.

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.

To configure session rates:

1. **max-burst-rate**—Enter a number to set how many SIP session invitations or H.323 SETUPs this session agent can send or receive (per second) within the configured burst rate window value. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum— $2^{32} - 1$

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.
2. **max-inbound-burst-rate**—Enter the maximum burst rate (number of session invitations per second) for inbound sessions from this session agent. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
3. **max-outbound-burst-rate**—Enter the maximum burst rate (number of session invitations per second) for outbound sessions to this session agent. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
4. **max-sustain-rate**—Enter a number to set the maximum rate of session invitations (per second) this session agent can send or receive within the current window. The default value is zero (0). The valid range is:
 - Minimum—zero (0)
 - Maximum— $2^{32} - 1$

The number you enter here must be larger than the number you enter for **max-burst-rate**.

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.

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.

5. **max-inbound-sustain-rate**—Enter the maximum sustain rate (of session invitations allowed within the current window) of inbound sessions from this session agent. This value should be larger than the **max-inbound-burst-rate** value. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
6. **max-outbound-sustain-rate**—Enter the maximum sustain rate (of session invitations allowed within the current window) of outbound sessions to this session agent. This value should be larger than the **max-outbound-burst-rate** value. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
7. **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 default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum— $2^{32} - 1$
8. **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). The valid range is:
 - Minimum—0
 - Maximum— $2^{32} - 1$

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. Other session agent parameters have been omitted for brevity.

session-agent	
constraints	enabled
max-sessions	355
max-inbound-sessions	355
max-outbound-sessions	355

The following example shows session agent constraints are enabled and the session rate parameters have been configured. Other session agent parameters have been omitted for brevity.

session-agent	
max-burst-rate	0
max-inbound-burst-rate	10
max-outbound-burst-rate	1
max-sustain-rate	3000
max-inbound-sustain-rate	0
max-outbound-sustain-rate	0
burst-rate-window	0

sustai n-rate-wi ndow

0

Configuring Realm Bandwidth

To configure admission control based on bandwidth, you set the max and min bandwidth parameters in the realm configuration.

To configure realm bandwidth:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **confi gure termi nal**
2. Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **medi a-manager**
3. Type **realm-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(medi a-manager)# **real m-confi g**
ACMEPACKET(real m-confi g)#
 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. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum— $2^{32} - 1$
 - 4b. The following example shows the maximum bandwidth for the realm has been configured. All other realm parameters have been omitted for brevity.

```
real m-confi g
      max-bandwi dth          64000
```

SIP Admission Control

You can configure the registered endpoint to accept and process requests from SIP realms. If a request does not meet the criteria of the option you choose here, it is rejected with a 403 (Forbidden) response.

To configure admission control:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **confi gure termi nal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **sessi on-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)# **si p-i nterface**
ACMEPACKET(si p-i nterface)#
 4. Type **sip-ports** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(si p-i nterface)# **si p-port**
ACMEPACKET(si p-port)#
 5. Set the criteria for admission control.
 - 5a. **allow-anonymous**—Enter the anonymous connection mode you want applied when SIP requests are processed. The default value is **all**.
The following are valid values:

- **all**—No ACL is applied and all anonymous connections are allowed.
- **agents-only**—Only requests from configured session agents are processed. The Net-Net SBC responds to all other requests with a *forbidden* response.
- **realm-prefix**—Only requests from session agents and addresses matching the realm's address prefix are processed. All other requests are rejected with a 403 (Forbidden) response.
- **registered**—Only requests from session agents and registered endpoints are processed. REGISTER allowed from any endpoint.
- **registered-prefix**—Only requests from session agent and registered endpoint addresses that match the realm's realm prefix are processed.

The following example shows the **allow-anonymous** parameter that has been configured to allow only requests from session agents and registered endpoints. All other session agent parameters following the **allow-anonymous** parameters are omitted for brevity.

```
si p-port
      address
      port          5060
      transport-protocol   UDP
      allow-anonymous    registered
```

H.323 Admission Control

You can configure the endpoint to allow accept and process requests from a H.323 realm. If a request does not meet the criteria you set here, it is rejected.

To configure admission control:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter>.
ACMEPACKET(config)# session-router
3. Type **h323** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# h323
ACMEPACKET(h323)#
4. Type **h323-stacks** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(h323)# h323-stacks
ACMEPACKET(h323-stack)#
5. Set the criteria upon which you want to base admission control.
 - 5a. **allow-anonymous**—Enter the anonymous connection option (mode) you want applied to the processing of H.323 requests. The default value is **all**.
 The following are valid values:
 - **all**—No ACL is applied and all anonymous connections are allowed.
 - **agents-only**—Only requests from configured session agents are processed.
 - **realm-prefix**—Only requests from session agents and addresses matching the realm's address prefix are processed. All other requests are rejected.

The following example shows the **allow-anonymous** parameter has been configured to allow only requests from configured session agents. All other **h.323-stack** parameters are omitted for brevity.

```
h323-stack
    allow-anonymous
        agents-only
```

MGCP Nested Realms

The Net-Net SBC can perform admission control based on the realm prefix and uses the bandwidth defined in the realm where a particular endpoint resides. At boot-up, the Net-Net SBC loads all realms that use a specific MGCP configuration for signaling, and it loads all realms and their corresponding address prefixes for that MGCP configuration.

After the Net-Net SBC receives a NTFY message from an endpoint, it decides in which child realm the endpoint resides and stores that realm in the information corresponding to the endpoint. When the Net-Net SBC needs to setup the media for that endpoint, it uses that information to decide bandwidth and steering port allocation for the realm—or for its parent, depending on your configuration.

To configure MGCP nested realms:

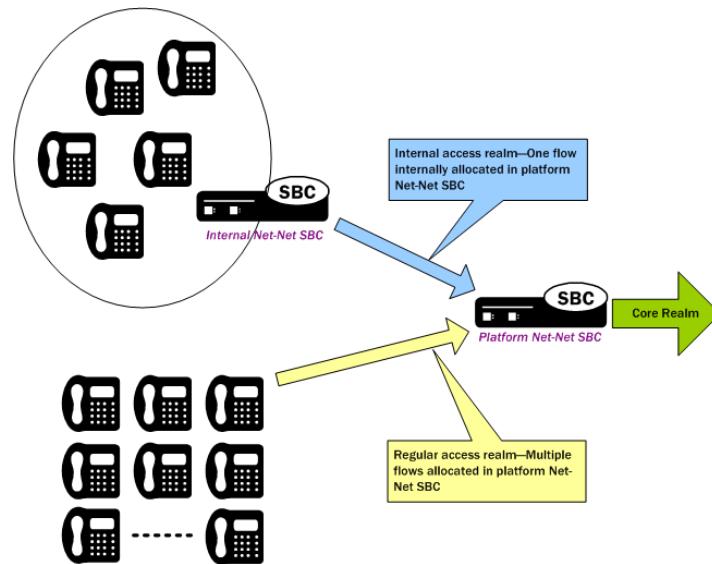
1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the **session-router** path.
 ACMEPACKET(**configure**)# **session-router**
3. Type **mgcp-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
 ACMEPACKET(**session-router**)# **mgcp-config**
 ACMEPACKET(**mgcp-config**)#
4. You can either add support to a new MGCP configuration or to an existing MGCP configuration:
 - 4a. If you do not currently have an MGCP configuration, you can add the option by typing options, a <Space> and then **nested-realm**.
 ACMEPACKET(**mgcp-config**)# **options nested-realm**
 - 4b. Select the MGCP configuration so that you can add MGCP nested realm support to it. Then, to add this option to a list of options that you have already configured for the MGCP configuration, type **options** followed by a <Space>, the plus sign (+), and the **nested-realm** option.
 ACMEPACKET(**mgcp-config**)# **select**
 ACMEPACKET(**mgcp-config**)# **options +nested-realm**
5. Save your work using the ACLI **save** or **done** command.

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, H.323, and MGCP. 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# config terminal
ACMEPACKET(config)#

```

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

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

```

3. Type **access-control** and press <Enter>.

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

```

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **realm-id**—Enter the name of a valid realm.
5. **application-protocol**—Enter a valid application protocol. There is no default for this parameter, and valid values are: **SIP**, **H.323**, or **MGCP**.
6. **access**—Set this parameter to **permit** (default).
7. **trust-level**—Set this parameter to **high**, changing it from the default (**none**).
8. **minimum-reserved-bandwidth**—Enter the minimum reserved bandwidth you want for the session agent, and that will trigger the creation of a separate pipe for it. Only a non-zero value will allow the feature to work properly, along with the other required values set out in these instructions. The default is 0, and the maximum is 0xffffffff (or 4294967295).
9. Save and activate your configuration.

Aggregate Session Constraints for SIP

You can set a full suite of session constraints and then apply them to a SIP interface. The session constraints configuration contains many of the same parameters as the session agent, so you can configure a group of constraints and then apply them to a SIP interface/

The SIP interface configuration's **constraint-name** parameter invokes the session constraint configuration you want to apply. Using the constraints you have set up, the 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.

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

Configuring Session Constraints

This section shows you how to configure aggregate session constraints and then apply them to a SIP interface.

The session constraints configuration contains many of the same parameters as the session agent does; it also incorporates the changes to the session agent parameters that are described in this section.

To configure the session constraints 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**—Enable this parameter to use these session constraints. The default value is **enabled**. The valid values are:
 - enabled | disabled
6. **max-sessions**—Enter the maximum sessions allowed for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999
7. **max-outbound-sessions**—Enter the maximum outbound sessions allowed for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999
8. **max-inbound-sessions**—Enter the maximum inbound sessions allowed for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999
9. **max-burst-rate**—Enter the maximum burst rate (invites per second) allowed for this constraint. This value should be the sum of the **max-inbound-burst-rate** and the **max-outbound-burst-rate**. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999
10. **max-sustain-rate**—Enter the maximum rate of session invitations per second allowed within the current window for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999

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 default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999
12. **max-inbound-sustain-rate**—Enter the maximum inbound sustain rate (of session invitations allowed within the current window) for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—99999999

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 default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
14. **max-outbound-sustain-rate**—Enter the maximum outbound sustain rate (of session invitations allowed within the current window) for this constraint. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999

For the sustained rate, the 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. **time-to-resume**—Enter the number of seconds that is used to place an element (like a session agent) in the standby state when it has been taken out of service because of excessive transaction timeouts. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
16. **ttr-no-response**—Enter the time delay in seconds to wait before changing the status of an element (like a session agent) after it has been taken out of service because of excessive transaction timeouts. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
17. **in-service-period**—Enter the time in seconds that elapses before an element (like a session agent) can return to active service after being placed in the standby state. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
18. **burst-rate-window**—Enter the time in seconds that you want to use to measure the burst rate; the “window” is the time over which the burst rate is calculated, and is used for the overall burst rate as well as the inbound and outbound burst rates. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
19. **sustain-rate-window**—Enter the time in seconds used to measure the sustained rate; the “window” is the time over which the sustained rate is calculated, and is used for the overall sustained rate as well as the inbound and outbound sustained rates. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
20. 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# **config terminal**
2. Type **session-router** and press <Enter>
ACMEPACKET(config)# **session-router**
3. Type **sip-interface** and press <Enter>
ACMEPACKET(session-router)# **sip-interface**
4. **constraint-name**—Enter the name of the session constraints configuration that you want to apply to this SIP interface. There is no default for this parameter.
5. Save and activate your configuration.

Configuring CAC, Policing, and Marking for non-Audio, non-Video Media

Support for the AS Bandwidth Modifier

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.

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:

- In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
- Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#
- 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)#
- 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
- Save and activate your configuration.

To set any of the new media types in the media policy configuration:

- In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
- Type **media-manager** and press <Enter>.
ACMEPACKET(configure)# **media-manager**
ACMEPACKET(media-manager)#
- 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)#
- 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
- 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:

- In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
- Type **session-router** and press <Enter>.
ACMEPACKET(configure)# **session-router**
ACMEPACKET(session-router)#
- 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) #

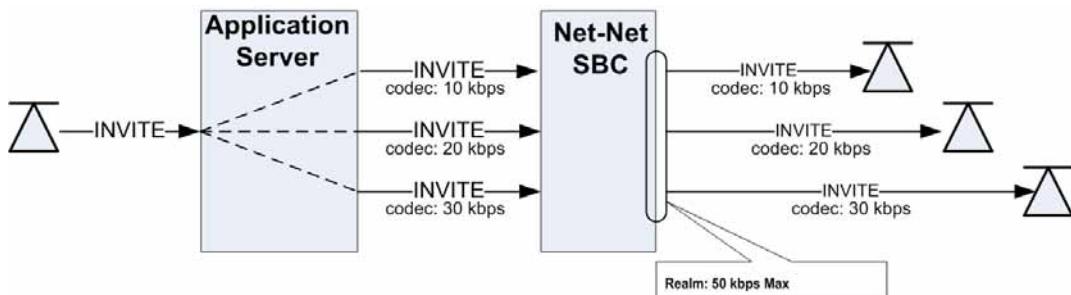
4. **sdp-bandwidth**—Enable this parameter to use the AS bandwidth modifier in the SDP in the conditions described in the [Support for the AS Bandwidth Modifier \(837\)](#) section above. The default is **disabled**. Valid values are:
 - enabled | disabled
5. **sdp-rate-limit-headroom**—Specify the percentage 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.

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.

CALL A					
CALL B		Ingress Realm A	Egress Realm B	Ingress Realm C	Egress Realm D
	Ingress Realm A	bandwidth shared	N/A	bandwidth not shared	N/A
	Egress Realm B	N/A	bandwidth shared	N/A	bandwidth not shared
	Ingress Realm C	bandwidth not shared	N/A	bandwidth shared	N/A
	Egress Realm D	N/A	bandwidth not shared	N/A	bandwidth shared

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.

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-profile** and press <Enter>.

```
ACMEPACKET(session-router)# sip-profile
ACMEPACKET(sip-profile)#
```
4. **name**—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's **name** when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. **forked-cac-bw**—Set this parameter to **shared** if you want forked sessions to share bandwidth resources, or set it to **per-session** if you want bandwidth to be counted for each session individually. There is no default for this parameter, and leaving it blank means:
 - For an ingress session agent without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Net-Net SBC will reference the associated realm.
 - For an ingress realm without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Net-Net SBC will reference the associated SIP interface.

- For an ingress SIP interface without a SIP profile or with a SIP profile where the forked CAC mode is blank, the Net-Net SBC will not perform bandwidth sharing for forked calls.
6. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- realm-config
- sip-interface
- session-agent

To apply a SIP profile to a SIP interface:

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

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

```

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

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

```

3. Type **sip-interface** and press <Enter>.

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

```

4. **sip-profile**—Enter the name of SIP profile configuration that includes the forked-cac-bandwidth parameter configured.

5. Save your work.

RADIUS Accounting Support

VSA 171, Acme-Session-Forked-Call-Id, is part of the Acme Packet RADIUS dictionary. The VSA is a string value, and appears as the header-value without the header parameters from the P-Multiring-Correlator header for a session identified as part of a forked call.

Monitoring

Using the ACLI **show sipd forked** command, you can display the total number of forked sessions the Net-Net SBC received and the total number it rejected. The Net-Net SBC counts forked sessions when it receives a dialog-creating INVITE and is enabled to shared bandwidth. Further, it counts as forked all session with the P-Multiring-Correlator header.

```
ACMEPACKET# show sipd forked
11:19:20-116
Forked Sessions          ----- Lifetime -----
                           Recent   Total   PerMax
Forked Sessions           0        0      0
Forked Sessions Rej       0        0      0
```

Conditional Bandwidth CAC for Media Release

The 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**

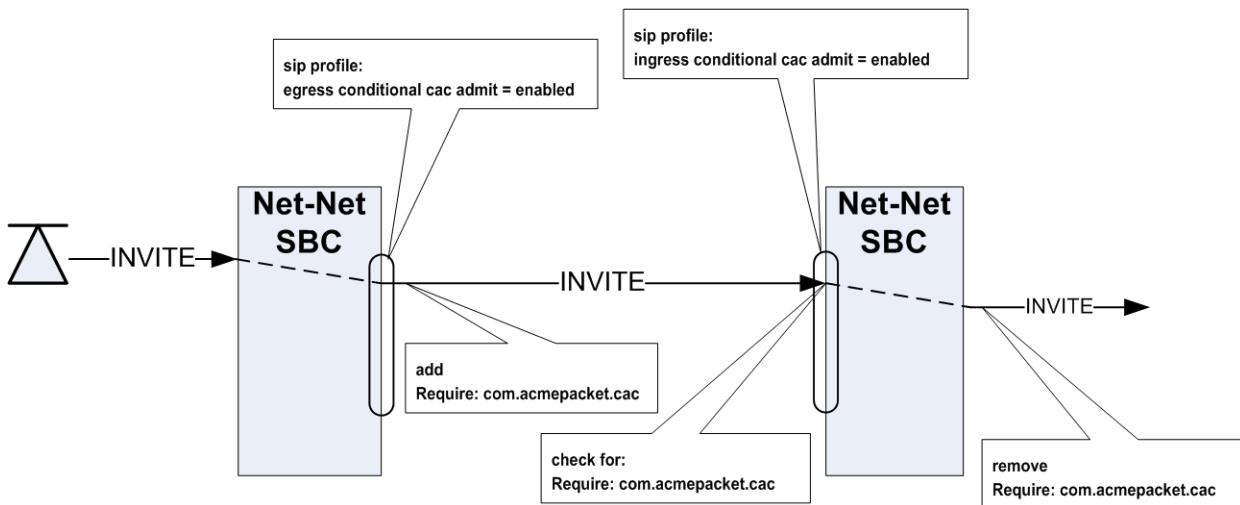
and the following parameter is set to enabled:

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

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

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

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

```
ACMEPACKET(sip-profile)#

```

4. **name**—Enter a name for this SIP profile configuration. This parameter is blank by default, and it is required. You will need the SIP profile's **name** when you want to apply this profile to a realm, SIP interface, or SIP session agent.
5. **ingress-conditional-cac-admit**—Set this parameter to enabled to process an INVITE with a Require tag as received on an ingress interface. You can set this parameter to disabled if you do not want to use this feature on the ingress side. There is no default for this parameter.
6. **egress-conditional-cac-admit**—Set this parameter to enabled if you want to use conditional bandwidth CAC for media release for calls that are first received by this system. This results in option tags being inserted on the INVITE's egress if the conditional CAC conditions are met. You can set this parameter to disabled if you do not want to use this feature. There is no default for this parameter.
7. Save your work.

Applying a SIP Profile

Once you have configured one or more SIP profiles, you can apply them to realms, SIP interfaces, and SIP session agents. As an example, this section shows you how to apply a SIP profile to a SIP interface. But the parameter name is the same in these configurations:

- `realm-config`
- `sip-interface`
- `session-agent`

To apply a SIP profile to a realm:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#

```
2. Type `session-router` and press <Enter>.

```
ACMEPACKET(config)# 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# config terminal

```

2. Type **session-router** and press <Enter> to access the **session-router** path.
ACMEPACKET(configure)# **session-router**
3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **sip-config**
4. Use the ACLI **select** command so that you can work with the SIP configuration.
ACMEPACKET(sip-config)# **select**
5. **options**—Set the options parameter by typing **+options**, a <Space>, the option name **con-cac-tag=your-new-tag**, and then press <Enter>.
ACMEPACKET(sip-config)# **options +con-cac-tag=com.test.cac**
6. Save your work.

About QoS Reporting

This section describes the Net-Net SBC 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 SBC QoS reporting is a measurement tool that collects statistics on Voice over IP (VoIP) call flows for SIP and H.323. To provide information, the Net-Net SBC writes additional parameters to the Remote Authentication Dial-in User Service (RADIUS) call record.

You can use QoS statistics for SLA customer reporting, fault isolation, SLA verification, and traffic analysis. The Net-Net SBC 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 SBC, any time a call occurs a record is generated and sent. Only Stop RADIUS records contain the QoS statistic information.

Only RADIUS Stop records contain QoS information. For non-QoS calls, the attributes appear in the record, but their values are always be zero (0). When you review the list of QoS VSAs, please note that “calling” in the attribute name means the information is sent by the calling party and “called” in the attribute name means the information is sent by the called party.

For additional details about see the Net-Net SBC’s support for RADIUS, including a dictionary of the standard RADIUS attributes supported by the Net-Net SBC and the Acme Packet accounting VSAs, see the *Net-Net 4000 RADIUS Reference Guide*.

The following example shows a CDR that includes QoS data:

```

Wed Jun 13 18:26:42 2007
    Acct-Status-Type = Stop
    NAS-IP-Address = 127.0.0.100
    NAS-Port = 5060
    Acct-Session-Id = "SDgtu4401-
c587a3aba59dcae68ec76cb5e2c6fe6f-v3000i 1"
    Acme-Session-Ingress-CallId =
"8EDDDC21D3EC4A218FF41982146844310xac1ec85d"
    Acme-Session-Egress-CallId = "SDgtu4401-
c587a3aba59dcae68ec76cb5e2c6fe6f-v3000i 1"
    Acme-Session-Protocol-Type = "SIP"
    Calling-Station-Id = ""9998776565"
<sip:9998776565@10.10.170.2:5060>;tag=2ed75b8317f"
    Called-Station-Id = "<sip:7143221099@10.10.170.2:5060>"
    Acct-Terminate-Cause = User-Request
    Acct-Session-Time = 7
    h323-setup-time = "18:24:36.966 UTC JUN 13 2007"
    h323-connect-time = "18:24:37.483 UTC JUN 13 2007"
    h323-disconnect-time = "18:24:44.818 UTC JUN 13 2007"
    h323-disconnect-cause = "1"
    Acme-Session-Egress-Realm = "peer"
```

```
Acme-Session-Ingress-Real m = "core"
Acme-FlowID_FS1_F = "local host: 65544"
Acme-FlowType_FS1_F = "PCMA"
Acme-Flow-In-Real m_FS1_F = "core"
Acme-Flow-In-Src-Addr_FS1_F = 10.10.170.15
Acme-Flow-In-Src-Port_FS1_F = 49156
Acme-Flow-In-Dst-Addr_FS1_F = 10.10.170.2
Acme-Flow-In-Dst-Port_FS1_F = 31008
Acme-Flow-Out-Real m_FS1_F = "peer"
Acme-Flow-Out-Src-Addr_FS1_F = 10.10.130.2
Acme-Flow-Out-Src-Port_FS1_F = 21008
Acme-Flow-Out-Dst-Addr_FS1_F = 10.10.130.15
Acme-Flow-Out-Dst-Port_FS1_F = 5062
Acme-Calling-RTCP-Packets-Lost_FS1 = 0
Acme-Calling-RTCP-Avg-Jitter_FS1 = 15
Acme-Calling-RTCP-Avg-Latency_FS1 = 0
Acme-Calling-RTCP-MaxJitter_FS1 = 15
Acme-Calling-RTCP-MaxLatency_FS1 = 0
Acme-Calling-RTP-Packets-Lost_FS1 = 0
Acme-Calling-RTP-Avg-Jitter_FS1 = 3
Acme-Calling-RTP-MaxJitter_FS1 = 44
Acme-Calling-Octets_FS1 = 957
Acme-Calling-Packets_FS1 = 11
Acme-FlowID_FS1_R = "local host: 65545"
Acme-FlowType_FS1_R = "PCMA"
Acme-Flow-In-Real m_FS1_R = "peer"
Acme-Flow-In-Src-Addr_FS1_R = 10.10.130.15
Acme-Flow-In-Src-Port_FS1_R = 5062
Acme-Flow-In-Dst-Addr_FS1_R = 10.10.130.2
Acme-Flow-In-Dst-Port_FS1_R = 21008
Acme-Flow-Out-Real m_FS1_R = "core"
Acme-Flow-Out-Src-Addr_FS1_R = 10.10.170.2
Acme-Flow-Out-Src-Port_FS1_R = 31008
Acme-Flow-Out-Dst-Addr_FS1_R = 10.10.170.15
Acme-Flow-Out-Dst-Port_FS1_R = 49156
Acme-Called-RTCP-Packets-Lost_FS1 = 0
Acme-Called-RTCP-Avg-Jitter_FS1 = 13
Acme-Called-RTCP-Avg-Latency_FS1 = 0
Acme-Called-RTCP-MaxJitter_FS1 = 21
Acme-Called-RTCP-MaxLatency_FS1 = 0
Acme-Called-RTP-Packets-Lost_FS1 = 0
Acme-Called-RTP-Avg-Jitter_FS1 = 0
Acme-Called-RTP-MaxJitter_FS1 = 3
Acme-Called-Octets_FS1 = 77892
Acme-Called-Packets_FS1 = 361
Acme-Firmware-Version = "C5.0.0"
```

```

Acme-Local-Time-Zone = "Time Zone Not Set"
Acme-Post-Dial-Delay = 110
Acme-Primary-Routing-Number =
"sip:7143221099@10.10.170.2:5060"
Acme-Ingress-Local-Addr = "10.10.170.2:5060"
Acme-Ingress-Remote-Addr = "10.10.170.15:5060"
Acme-Egress-Local-Addr = "10.10.130.2:5060"
Acme-Egress-Remote-Addr = "10.10.130.15:5060"
Acme-Session-Disposition = 3
Acme-Di disconnect-Initiator = 2
Acme-Di disconnect-Cause = 16
Acme-SIP-Status = 200
Acme-Egress-Final-Routing-Number =
"sip:7143221099@10.10.130.15:5060"
Acme-CDR-Sequence-Number = 14
Client-IP-Address = 172.30.20.150
Acct-Unique-Session-Id = "0832b03cd3a290b3"
Timestamp = 1181773602

```

Configuring QoS

This section explains how to configure QoS. To generate QoS metrics, you need to enable QoS for the realm of the originating caller. The ingress realm determines whether QoS is turned on for a specific flow.

Note: If you run with QoS turned on one side only and disabled on the other you lose the ability to measure latency through the use of RTCP timestamps.

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(real-m-config)#
4. **qos-enable**—Enable this parameter. The default value is **disabled**.

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 and how to set them appear in the instructions and examples section.

When a SIP INVITE or an H.323 Setup 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 (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** parameter) 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

With Release 6.0, the Net-Net SBC now normalizes the user-part of the Request-URI prior to performing any matching for NMC based on the dialed number. A departure from this feature's prior implementation, this normalization strips out any of the visual-separator characters.

Note that normalization occurs only for NMC look-up purposes, and it does not alter the actual Request-URI. For previous releases, NMC rule matching based on the dialed number fails when the dialed number has visual separators or additional parameters such as: rn, npdi, ci c, postd, etc. 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 would be normalized to #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.

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 treatment method is call diversion, the Net-Net SBC sends the call to its SIP or H.323 task for completion, and this task does not repeat network management control rule application.

Before sending on the call to its SIP or H.323 task, the Net-Net SBC adds the acme_i wf_nmc=<nmc_name> Request URI parameter to the Request URI (where nmc_name is the name of a control rule). When the INVITE is received and that parameter is found, the Net-Net SBC applies the specified control rule to the session.

- 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 message 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
 - For H.323, the Net-Net SBC sends a releaseComplete message with the Q.850 cause code (that you configure as part of the control rule) of the control rule as the Q.931 Cause IE.
- 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. DNS queries only apply to pure SIP or IWF calls that originate in H.323 and are interworked to SIP.
- 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.

Priority Call Exemption from Policy Server Approval

The Net-Net SBC now identifies priority calls and provides expedited treatment for them, even if these calls use associated realms for which there is an associated policy server handling bandwidth allocation. Instead of waiting for a response from the policy server, the Net-Net SBC immediately processes the call. When and if the policy server responds, the Net-Net SBC handles the response, but in all likelihood the priority calls have already been processed.

Enhanced Call Gapping

NMC provides flexibility by allowing a desired call-per-second (CPS) threshold to be achieved or surpassed by a predictable amount. Referred to as “call gapping,” this allows the Net-Net SBC to average the call rate and widen the period of a surge that would invoke NMC rules.

Without call gapping enabled, 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 $1/\text{gap-rate}$ 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 uses two 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.

About the Call Gapping Algorithm

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:

- $1 + m - m * A * \text{del taT/W} \leq M$ —Means the call is allowed
- $1 + m - m * A * \text{del taT/W} > M$ —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 it uses when the relevant parameters are not configured.

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# config terminal
2. Type **session-router** and press <Enter> to access the signaling-related configurations.
ACMEPACKET(config)# 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**—Enable or disable this network management control rule. The default value is **enabled**. The valid values are:

- enabled | disabled

6. **destination-identifier**—Enter the call 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 (^).
 You can enter special characters in the **destination-identifier** parameter. You can enter 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), lowercase letter p (p), lowercase letter w (w).

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. The valid values are:
 - **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. Use this value if you want to enable [Priority Call Exemption from Policy Server Approval \(852\)](#)

For more information about control types, refer to the [Call Handling Determination \(851\)](#) 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 value is zero (0). The valid values are:
 - **GAP-RATE**—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

- **GAP-PERCENT**—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 you want to use. The default value is **none**. The valid values are:
- **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 \(852\)](#) section above.
10. **next-hop**—Enter the next hop for the Net-Net SBC to use when the treatment method is **DIVERT**. The valid values are:
- 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**. The default value is **503** (Service Unavailable). The valid range is:
- Minimum—1
 - Maximum—699
14. **cause-code**—Enter the Q.850 cause code that you want the Net-Net SBC to use when the treatment method is **REJECT**. The default value is **63** (Service or option not available). The valid range is:
- Minimum—1
 - Maximum—999999999
- For a SIP call, the Net-Net SBC replaces the cause code in the Reason header of the SIP response.
- For a H.323 call, the Net-Net SBC converts the cause code to a Q.931 cause code in the Q.931 Cause IE in the releaseComplete message.

Enabling Enhanced Call Gapping

Enhanced NMC call gapping uses 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. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
- **gap-rate-window-size**—Length of time in seconds used for the gapping rate calculation. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999

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# config 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.
- **gap-rate-max-count**—Maximum count that triggers the application of network management control rule if it is exceeded. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
 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.
- **gap-rate-window-size**—Length of time in seconds used for the gapping rate calculation. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
 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.
4. Save and activate your configuration.

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# **config terminal**
2. Type **media-manager** and press <Enter>.
ACMEPACKET(config)# **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(real m-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.

Accounting Configuration for QoS

This section explains how to configure the account configuration and account servers so you can use the Net-Net SBC 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# **config terminal**
2. Type **session-router** and press <Enter>.
ACMEPACKET(config)# **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
                                         Remote-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 SBC in hostname (FQDN) format. The default value is the name of the local host. The value you enter here must match the configured physical interface's operation type **control** or **maintenance**, to determine on which network to send RADIUS messages.
2. **port**—Enter the number of the UDP port associated with the Net-Net SBC from which RADIUS messages are sent. The default value is **1813**. The valid range is:
 - Minimum—1025
 - Maximum—65535
3. **strategy**—Indicate the strategy you want used to select the accounting servers to which the Net-Net SBC will send its accounting messages. The default value is **hunt**. The following table lists the available strategies:
 - **hunt**—Selects accounting servers in the order in which they are listed. If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers
 - **failover**—Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
 - **roundrobin**—Selects each accounting server in order, distributing the selection of each accounting server evenly over time.

- **fastestrtt**—Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
 - **fewestpending**—Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the Net-Net SBC).
4. **state**—Enable this parameter if you want the account configuration active on the system. Disable it if you do not want the account configuration active on the system. The default value is **enabled**. The valid values are:
- enabled | disabled
5. **max-msg-delay**—Indicate the length of time in seconds that you want the Net-Net SBC to continue trying to send each accounting message. During this delay, the Net-Net SBC can hold a generic queue of 4096 messages. The default value is **60**.
- Minimum—zero (0)
 - Maximum— $2^{32}-1$
6. **max-wait-failover**—Indicate the maximum number of accounting messages the Net-Net SBC 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 SBC attempts to send its accounting messages, including its pending messages, to the next accounting server in its configured list. The default value is **100**. The valid range is:
- Minimum—1
 - Maximum—4096
7. **trans-at-close**—Disable this parameter if you do not want to defer the transmission of message information to the close of a session. Enable it if you want to defer message transmission. The default value is **disabled**. The valid values are:
- **disabled**—The Net-Net SBC 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 SBC used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.
8. **generate-start**—Select the type of SIP event that triggers the Net-Net SBC to transmit a RADIUS Start message. The default value is **ok**. The valid values are:
- **start**—RADIUS Start message should not be generated
 - **invite**—RADIUS Start message should be generated once the Net-Net SBC receives a SIP session INVITE.
 - **ok**—RADIUS Start message is generated once the Net-Net SBC receives an OK message in response to an INVITE.
9. **generate-interim**—Retain the default value **reinvite-response** to cause the Net-Net SBC 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:

- **ok**—RADIUS Start message is generated when the Net-Net SBC receives an OK message in response to an INVITE.
 - **reinvite**—RADIUS Interim message is generated when the Net-Net SBC receives a SIP session reINVITE message.
 - **reinvite-response**—RADIUS Interim message is generated when the Net-Net SBC 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 SBC receives a SIP session reINVITE, and the Reinvite is cancelled before the Net-Net SBC 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 SBC 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**—Enter the number of the UDP port associated with the account server to which RADIUS messages are sent. The default value is **1813**. The valid range is:
 - Minimum—1025
 - Maximum—65535
3. **state**—Enable or disable the account servers on the system. The default value is **enabled**. The valid values are:
 - enabled | disabled
4. **min-round-trip**—Indicate the minimum round trip time of an accounting message in milliseconds. The default value is **250**. The valid range is:
 - Minimum—10
 - Maximum—5000

A round trip consists of the following:

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

If the fastest RTT is the strategy for the account configuration, the value you enter here can be used to determine an order of preference (if all the configured account servers are responding in less than their minimum RTT).
5. **max-inactivity**—Indicate the length of time in seconds that you want the Net-Net SBC with pending accounting messages to wait when it has not received a valid response from the target account server. The default value is **60**. The valid range is:
 - Minimum—1

- Maximum—300

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

6. **restart-delay**—Indicate the length of time in seconds you want the Net-Net SBC to wait before resending messages to a disabled account server. The default value is 30. The valid range is:

- Minimum—1
- Maximum—300

7. **bundle-vsa**—Retain the default **enabled** if you want the account server to bundle the VSAs within RADIUS accounting messages. Enter **disabled** if you do not want the VSAs to be bundled. (Bundling means including multiple VSAs within the vendor value portion of the message.) The valid values are:

- enabled | disabled

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

8. **secret**—Enter the secret passed from the account server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IPv4 address of the received packet.
9. **NAS-ID**—Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the Net-Net SBC 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 SBC accounting messages always includes in the first of these:

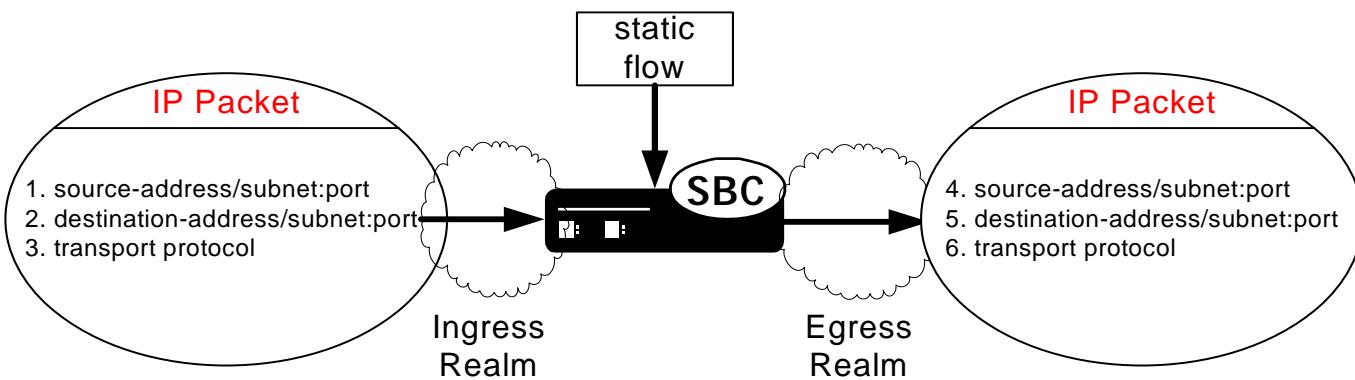
- Network Access Server (NAS) IP address (the IP address of the Net-Net SBC's SIP proxy)
- NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

Introduction

This chapter describes the Net-Net SBC's static flows feature. Static flows allow network traffic that matches specific criteria to pass through the Net-Net SBC 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 SBC 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 SBC 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 SBC does the following:

1. The IPv4 traffic is forwarded out of the Net-Net SBC 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

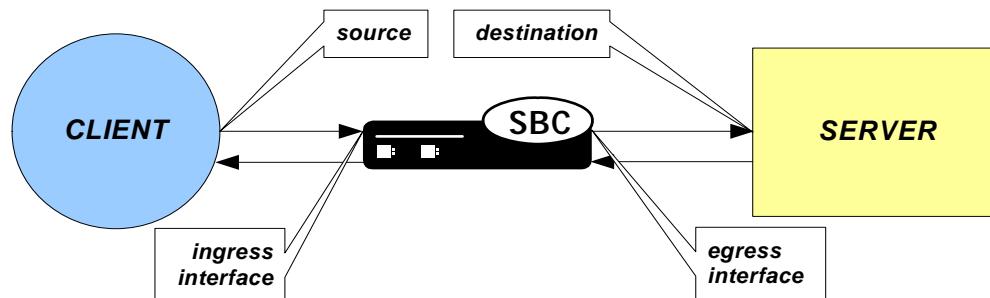
The Net-Net SBC supports Network Address and Port Translation (NAPT) and Trivial File Transfer Protocol (TFTP) functionality over media interfaces, collectively known as Network Address Translation (NAT) ALG. The NAT ALG feature is implemented as an extension of the static flow feature.

In some applications, the Net-Net SBC acts as an intermediary device, positioned between endpoints located in an access network and application servers located in a backbone network. The Net-Net SBC's NAT ALG feature enables these endpoints to use non-VoIP protocols, such as TFTP and HTTP, to access servers in a provider's backbone network to obtain configuration information.

NAT ALG parameters support RTC and can be dynamically reconfigured. The active NAT ALG configuration can be replicated on the standby SD in an HA configuration.

NAPT

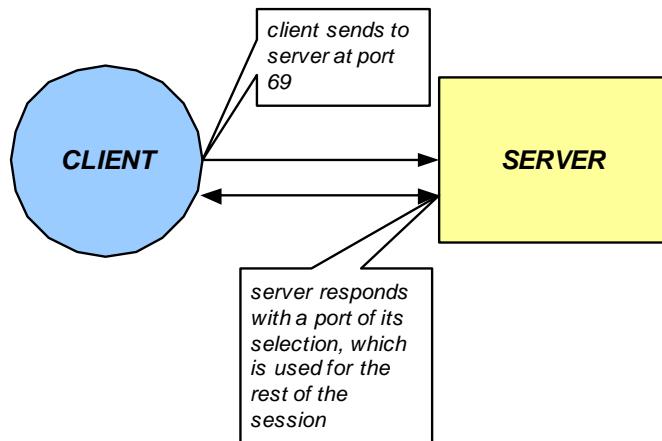
The NAPT ALG functionality is the same as that found in commercially available enterprise and residential NAT devices. The Net-Net SBC watches for packets entering a media interface that match source and destination IP address criteria. Matching packets are then redirected out of the egress interface, through a specified port range, toward a destination address.



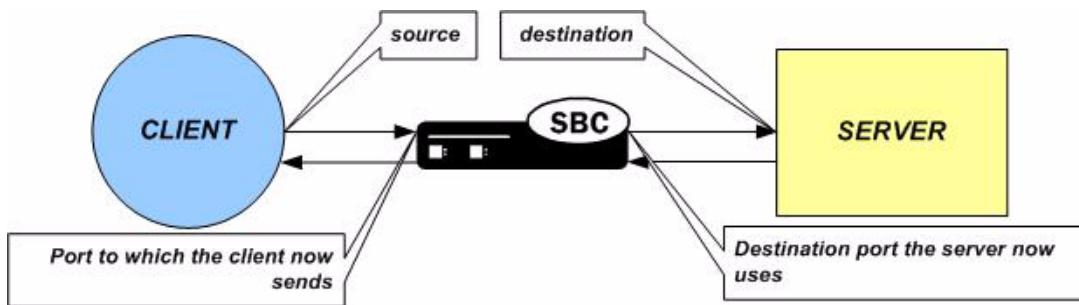
TFTP

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 SBC 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 SBC 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 SBC.
 2. Set the criteria that describes how traffic should be translated on the egress side of the Net-Net SBC. 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 SBC. 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 SBC. 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 SBC.

- **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 SBC 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 SBC 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 SBC where traffic exits the Net-Net SBC 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 SBC 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 SBC.
- **end-port**—The ending port the NAT ALG uses as it re-sources traffic on the egress side of the Net-Net SBC.

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# config terminal
2. Type **media-manager** and press <Enter> to access the **media-manager** path.
ACMEPACKET(configure)# media-manager
3. Type **static-flow** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# static-flow
From this point, you can configure media policing parameters.
4. **in-realm-id**—Enter the ingress realm or interface source of packets to match for static flow translation. This in-realm-id field value must correspond to a valid identifier field entry in a **realm-config**. This is a required field. Entries in this field must follow the Name Format.

5. **in-source**—Enter the incoming source IP address and port of packets to match for static flow translation. IP address of 0.0.0.0 matches any source address. Port 0 matches packets received on any port. The port value has no impact on system operation if either ICMP or ALL is the selected protocol. This parameter takes the format:

```
in-source <i p-address>[:<port>]
```

The default value is **0.0.0.0**. The valid port range is:

- Minimum—0
- Maximum—65535

6. **in-destination**—Enter the incoming destination IP address and port of packets to match for static-flow translation. An IP address of 0.0.0.0 matches any source address. Port 0 matches packets received on any port. The port value has no impact on system operation if either ICMP or ALL is the selected protocol. The in-source parameter takes the format:

```
in-destination <i p-address>[:<port>]
```

The default value is **0.0.0.0**. The valid port range is:

- Minimum—0
- Maximum—65535

7. **out-realm-id**—Enter the defined realm where traffic leaving this NAT ALG exits the Net-Net SBC.

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 SBC. Do not enter a port number for this parameter. The default value is **0.0.0.0**.

9. **out-destination**—Enter the IPv4 address and port number of the server or other destination to which traffic is directed. The default value is **0.0.0.0**. The valid port range is:

- Minimum—0
- Maximum—65535

10. **protocol**—Enter the protocol this NAPT ALG acts upon. The default value is **UDP**. The valid values are:

- TCP | UDP | ICMP | ALL

11. **alg-type**—Enter the type of NAT ALG to use. The default value is **none**. The valid values are:

- **none**—No dynamic ALG functionality
- **NAPT**—Configure as NAPT ALG
- **TFTP**—Configure as TFTP ALG

12. **start-port**—Enter the beginning port number of the port range that the Net-Net SBC allocates on the egress side for flows that this NAPT ALG redirects. The default value is **0**. The valid range is:

- Minimum—0, 1025
- Maximum—65535

13. **end-port**—Enter the ending port number of the port range that the Net-Net SBC allocates on the egress side for flows that this NAPT ALG redirects. The default value is **0**. The valid range is:

- Minimum—0, 1025
 - Maximum—65535
14. **flow-time-limit**—Enter the total time limit for a flow in seconds. A value of **0** means there is no limit. The valid range is:
- Minimum—0
 - Maximum—999999999
15. **initial-guard-timer**—Enter the initial guard timer value in seconds. A value of **0** means there is no limit. The valid range is:
- Minimum—0
 - Maximum—999999999
16. **subsq-guard-timer**—Enter the subsequent guard timer value in seconds. A value of **0** means there is no limit. The valid range is:
- Minimum—0
 - Maximum—999999999
17. **average-rate-limit**—Enter a maximum sustained rate limit in bytes per second. The default value is **0**; no limit. The valid range is:
- Minimum—0
 - Maximum—125000000

The following example shows a **static-flow** configuration element configured for a NAPT ALG.

in-real m-id	access
in-source	172.16.0.0/16
in-destination	172.16.1.16:23
out-real m-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

High Availability Nodes

Net-Net SBCs can be deployed in pairs to deliver high availability (HA). Two Net-Net SBCs operating in this way are called an HA node. Over the HA node, media and call state are shared, keeping sessions/calls from being dropped in the event of a failure.

Two Net-Net SBCs work together in an HA node, one in active mode and one in standby mode.

- The active Net-Net SBC checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it will hand over its role as the active system to the standby Net-Net SBC in the node.
- The standby Net-Net SBC is the backup system, fully synchronized with active Net-Net SBC's session status. The standby Net-Net SBC monitors the status of the active system so that, if needed, it can assume the active role without the active system having to instruct it to do so. If the standby system takes over the active role, it notifies network management using an SNMP trap.

In addition to providing instructions for how to configure HA nodes and their features, this chapter explains how to configure special parameters to support HA for all protocols.

Overview

To produce seamless switchovers from one Net-Net SBC to the other, the HA node uses shared virtual MAC and virtual IP addresses for the media interfaces in a way that is similar to VRRP (virtual router redundancy protocol). When there is a switchover, the standby Net-Net SBC sends out a gratuitous ARP messages using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted.

Within the HA node, the Net-Net SBCs advertise their current state and health to one another in checkpointing messages; each system is apprised of the other's status. Using Acme Packet's HA protocol, the Net-Net SBCs communicate with UDP messages sent out and received on the rear interfaces.

The standby Net-Net SBC shares virtual MAC and IPv4 addresses for the media interfaces (similar to VRRP) with the active Net-Net SBC. Sharing addresses eliminates the possibility that the MAC and IPv4 address set on one Net-Net SBC in an HA node will be a single point of failure. The standby Net-Net SBC sends ARP requests using a "utility" IPv4 address and its hard-coded MAC addresses to obtain Layer 2 bindings.

The standby Net-Net SBC assumes the active role when:

- It has not received a checkpoint message from the active Net-Net SBC for a certain period of time.

- It determines that the active Net-Net SBC's health score, a concept defined in this chapter's [Health Score \(872\)](#) section, has decreased to an unacceptable level.
- The active Net-Net SBC relinquishes the active role.

Establishing Active and Standby Roles

Net-Net SBCs establish active and standby roles in the following ways.

- If a Net-Net SBC boots up and is alone in the network, it is automatically the active system. If you then pair a second Net-Net SBC with the first to form an HA node, then the second system to boot up will establish itself as the standby automatically.
- If both Net-Net SBCs in the HA node boot up at the same time, they negotiate with each other for the active role. If both systems have perfect health, then the Net-Net SBC with the lowest HA rear interface IPv4 address will become the active Net-Net SBC. The Net-Net SBC with the higher HA rear interface IPv4 address will become the standby Net-Net SBC.
- If the rear physical link between the two Net-Net SBCs fails during boot up or operation, both will attempt to become the active Net-Net SBC. In this case, processing will not work properly.

Health Score

HA Nodes use health scores to determine their active and standby status. Health scores are based on a 100-point system. When a Net-Net SBC is functioning properly, its health score is 100.

Generally, the Net-Net SBC with the higher health score is active, and the Net-Net SBC with the lower health score is standby. However, the fact that you can configure health score thresholds builds some flexibility into using health scores to determine active and standby roles. This could mean, for example, that the active Net-Net SBC might have a health score lower than that of the standby Net-Net SBC, but a switchover will not take place because the active Net-Net SBC's health score is still above the threshold you configured.

Alarms are key in determining health score. Some alarms have specific health score value that are subtracted from the Net-Net SBC's health score when they occur. When alarms are cleared, the value is added back to the Net-Net SBC's health score.

You can look at a Net-Net SBC's health score using the ACLI **show health** command.

Switchovers

A switchover occurs when the active Net-Net SBC stops being the active system, and the standby Net-Net SBC takes over that function. There are two kinds switchovers: automatic and manual.

Automatic Switchovers

Automatic switchovers are triggered without immediate intervention on your part. Net-Net SBCs switch over automatically in the following circumstances:

- When the active Net-Net SBC's health score of drops below the threshold you configure.
- When a time-out occurs, meaning that the active Net-Net SBC has not sent checkpointing messages to the standby Net-Net SBC within the allotted time.

The active Net-Net SBC might not send checkpointing messages for various reasons such as link failure, communication loss, or advertisement loss. Even if the active Net-Net SBC has a perfect health score, it will give up the active role if it does not send a checkpoint message or otherwise advertise its status within the time-out window. Then the standby Net-Net SBC takes over as the active system.

When an automatic switchover happens, the Net-Net SBC that has just become active sends an ARP message to the switch. This message informs the switch to send future messages to its MAC address. The Net-Net SBC that has just become standby ignores any messages sent to it.

Manual Switchovers

You can trigger a manual switchover in the HA node by using the ACI **notify berpd force** command. This command forces the two Net-Net SBCs in the HA node to trade roles. The active system becomes standby, and the standby becomes active.

In order to perform a successful manual switchover, the following conditions must be met.

- The Net-Net SBC from which you trigger the switchover must be in one of the following states: active, standby, or becoming standby.
- A manual switchover to the active state is only allowed on a Net-Net SBC 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 Net-Net SBC in the standby or becoming standby state if it has a health score above the value you configure for the threshold.

State Transitions

Net-Net SBCs can experience series of states as they become active or become standby.

Note: Packet processing only occurs on an active Net-Net SBC.

State	Description
Initial	When the Net-Net SBC is booting.
Becoming Active	When the Net-Net SBC has negotiated to become the active system, but is waiting the time that you set to become fully active. Packets cannot be processed in this state.
Active	When the Net-Net SBC is handling all media, signaling, and configuration processing.
Relinquishing Active	When the Net-Net SBC is giving up its Active status, but before it has become standby. This state is very brief.
Becoming Standby	When the Net-Net SBC is becoming the standby system but is waiting to become fully synchronized. It remains in this state for the period of time you set in the becoming-standby-time parameter, or until it is fully synchronized.
Standby	When the Net-Net SBC is fully synchronized with its active system in the HA node.
OutOfService	When the Net-Net SBC cannot become synchronized in the period of time you set in the becoming-standby-time parameter.

State Transition Sequences

When the active Net-Net SBC assumes its role as the active system, but then changes roles with the standby Net-Net SBC to become standby, it goes through the following sequence of state transitions:

1. Active
2. RelinquishingActive
3. BecomingStandby
4. Standby

When the standby Net-Net SBC assumes its role as the standby system, but then changes roles with the active Net-Net SBC to become active, it goes through the following sequence of state transitions:

1. Standby
2. BecomingActive
3. Active

HA Features

HA nodes support configuration checkpointing, which you are required to set up so that the configurations across the HA node are synchronized. In addition, you can set up the following optional HA node features:

- Multiple rear interface support
- Gateway link failure detection and polling

Multiple Rear Interfaces

Configuring your HA node to support multiple rear interfaces eliminates the possibility that either of the rear interfaces you configure for HA support will become a single point of failure. Using this feature, you can configure individual Net-Net SBCs with multiple destinations on the two rear interfaces, creating an added layer of failover support.

When you configure your HA node for multiple rear interface support, you can use last two rear interfaces (wancom1 and wancom2) for HA—the first (wancom0) being used for Net-Net SBC management. You can connect your Net-Net SBCs using any combination of wancom1 and wancom2 on both systems. Over these rear interfaces, the Net-Net SBCs in the HA node share the following information:

- Health
- Media flow
- Signaling
- Configuration

For example, if one of the rear interface cables is disconnected or if the interface connection fails for some other reason, all health, media flow, signaling, and configuration information can be checkpointed over the other interface.

Health information is checkpointed across all configured interfaces. However, media flow, signaling, and configuration information is checkpointed across one interface at a time, as determined by the Net-Net SBC's system HA processes.

Configuration Checkpointing

During configuration checkpointing, all configuration activity and changes on one Net-Net SBC are automatically mirrored on the other. Checkpointed transactions include adding, deleting, or modifying a configuration on the active Net-Net SBC. This means that you only need to perform configuration tasks on the active Net-Net SBC because the standby system will go through the checkpointing process and synchronize its configuration to reflect activity and changes.

Because of the way configuration checkpointing works, the ACI **save-config** and **activate-config** commands can only be used on the active Net-Net SBC.

- When you use the ACI **save-config** command on the active Net-Net SBC, the standby Net-Net SBC learns of the action and updates its own configuration. Then the standby Net-Net SBC saves the configuration automatically.
- When you use the ACI **activate-config** command on the active Net-Net SBC, the standby Net-Net SBC learns of the action and activates its own, updated configuration.

The ACI **acquire-config** command is used to copy configuration information from one Net-Net SBC to another.

Gateway Link Failure Detection and Polling

In an HA node, the Net-Net SBCs can poll for and detect media interface links to the gateways as they monitor ARP connectivity. The front gateway is assigned in the network interface configuration, and is where packets are forwarded out of the originator's LAN.

The Net-Net SBC monitors connectivity using ARP messages that it exchanges with the gateway. The Net-Net SBC sends regular ARP messages to the gateway in order to show that it is still in service; this is referred to as a "heartbeat" message. If the Net-Net SBC deems the gateway unreachable for any of the reasons discussed in this section, a network-level alarm is generated and an amount you configure for this fault is subtracted from the system's health score.

The Net-Net SBC generates a "gateway unreachable" network-level alarm if the Net-Net SBC has not received a message from the media interface gateway within the time you configure for a heartbeat timeout. In this case, The Net-Net SBC will send out ARP requests and wait for a reply. If no reply is received after resending the set number of ARP requests, the alarm remains until you clear it. The health score also stays at its reduced amount until you clear the alarm.

When valid ARP requests are once again received, the alarm is cleared and system health scores are increased the appropriate amount.

You can configure media interface detection and polling either on a global basis in the SD HA nodes/redundancy configuration or on individual basis for each network interface in the network interface configuration.

Before You Configure

Before configuring the parameters that support HA, be sure that you have completed the following steps.

1. Set up physical connections between the Net-Net SBCs. For more information, refer to this chapter's [HA Node Connections \(877\)](#) section.
 - Avoid breaking the physical link (over the rear interfaces) between the Net-Net SBCs in an HA node once you have established that link, configured the active Net-Net SBC, and acquired that configuration on the standby Net-Net SBC. If the physical link between the Net-Net SBCs breaks, they will both attempt to become the active system and HA will not function as designed.

2. Confirm that both Net-Net SBCs are set to the same time. Use the ACCLI **show clock** command to view the system time. If the Net-Net SBCs show different times, use the **system-timeset** command to change it.

We recommend that you use NTP to synchronize your Net-Net SBCs so that they have a common stratum time source.

3. HA nodes use ports 1 and 2 as the HA interfaces. As a rule, set port 0 on the rear panel of the Net-Net SBC chassis as the boot and management interface. You configure all rear interfaces in the physical interface configuration.
4. For ACCLI configuration, you will need to know the target names of the Net-Net SBCs making up the HA node. The target name of the system is reflected in the ACCLI's system prompt. For example, in the **ACMEPACKET#** system prompt, **ACMEPACKET** is the target name.

You can also see and set the target name in the Net-Net SBC's boot parameters. For more information about boot parameters, refer to this guide's *Getting Started* chapter.

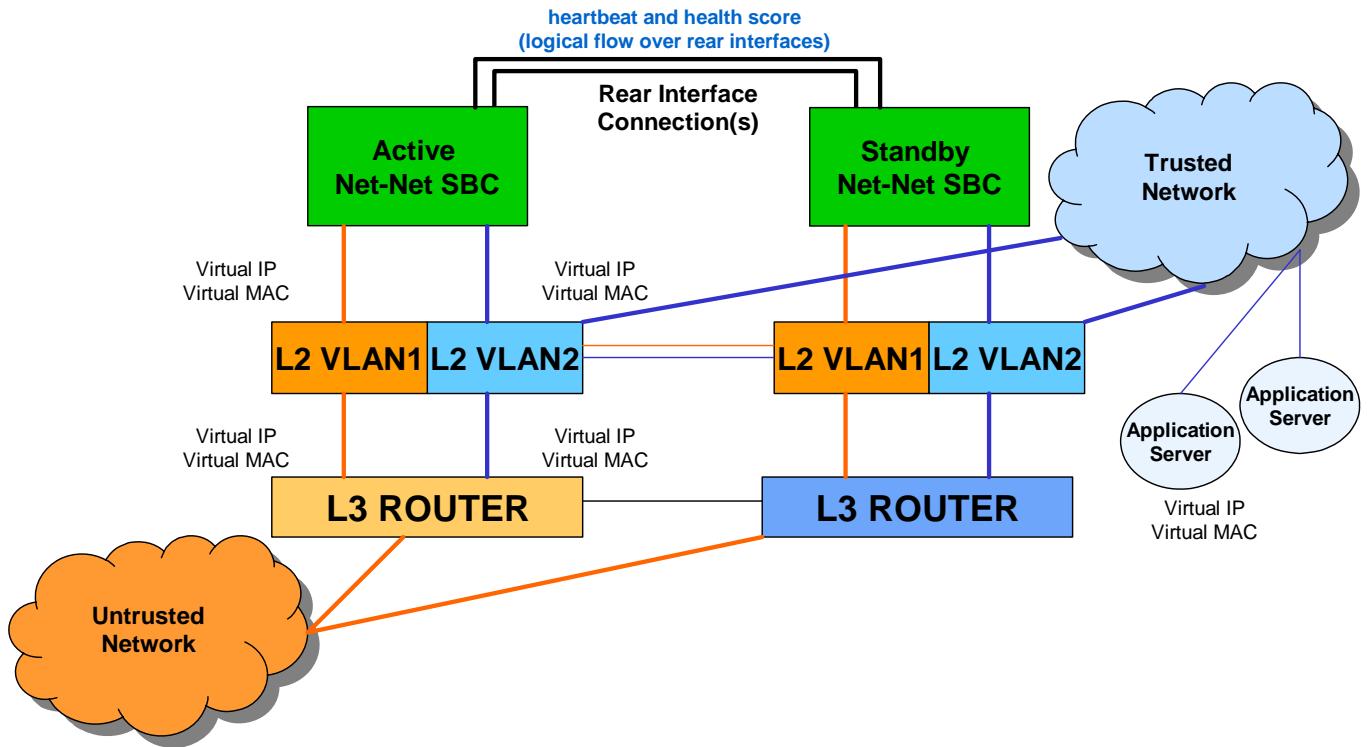
The target name is case sensitive. Note these values with care so that you can enter the correct, case sensitive target name.

5. Devise virtual MAC addresses so that, if a switchover happens, existing sessions will not be interrupted. The MAC addresses that your HA node uses must be created according to the instructions in this chapter's [Virtual MAC Addresses \(879\)](#) section.

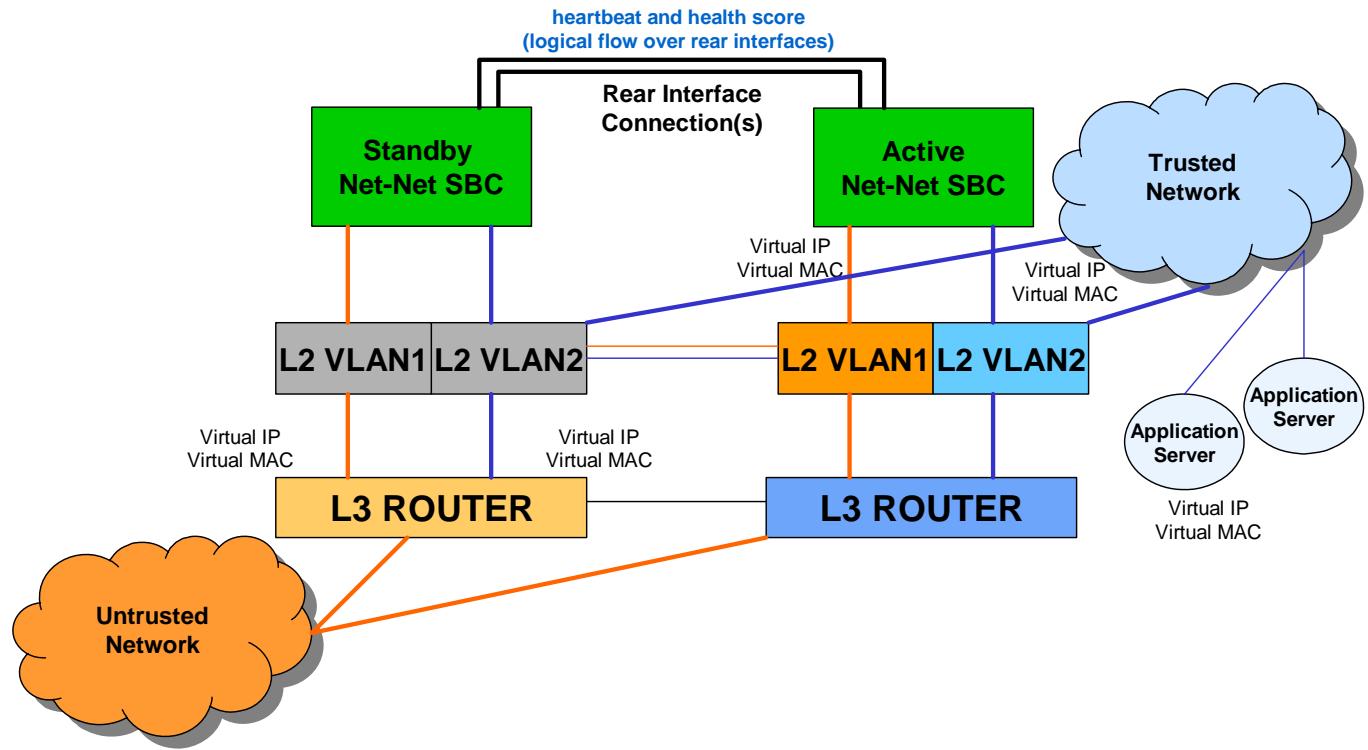
HA Node Connections

To use HA, you must establish Layer 2 and Layer 3 networks that interconnect two Net-Net SBCs and support HA with the required physical network connections. The basic network set-up in the following diagram shows an HA node deployment where each Net-Net SBC is connected to its own Layer 2 switch. This set-up provides a measure of added redundancy in the event that one of the switches fails.

Here, the active Net-Net SBC is using the virtual MAC and IP addresses.



In the second diagram, the same network is shown with the HA node having experienced a switchover. The previously standby Net-Net SBC has taken over the active role in the HA node and is using the virtual IP and MAC addresses.



Caution: Switches should never be in master-slave mode. If they are, HA will not work correctly.

The following are hardware set-up and location considerations for placing an HA Node:

- You must set up each Net-Net SBC according to the requirements and safety precautions set out in the *Net-Net System Hardware Installation Guide*.
- Each Net-Net SBC's media interfaces must be connected to the same switches (or other network entities), as shown in the diagram above.
- The length of the shielded crossover 10/100 category 5 Ethernet cable that connects the Net-Net SBCs from the rear interfaces must be able to reach from the configured rear interface on one Net-Net SBC to the configured rear interface on the other.

HA nodes use Acme Packet's border element redundancy protocol for its tasks. This protocol uses a connection between the rear interfaces of two Net-Net SBCs to checkpoint the following information: health, state, media flow, signaling, and configuration.

Caution: We recommend that you use shielded category 5 (RJ45) crossover cables for all 10/100 Ethernet connections used for HA.

You can set up either single or multiple rear interface support for your HA node. For single interface support, one cable connects the two Net-Net SBCs; for multiple interface support, two cables are used. However, the software configurations for each type of connection mode are different; steps for each are provided in this chapter's [Configuring HA Node Connections \(882\)](#) section.

Note: When you make these connections, do not use port 0 (wancom0) on the rear interface of the Net-Net SBC chassis; that port should only be used for Net-Net SBC management. Instead, use ports 1 and 2 (wancom1 and wancom2).

To cable Net-Net SBCs using single rear interface support:

1. Using a 10/100 category 5 crossover cable, insert one end into either port 1 (wancom1) or port 2 (wancom2) on the rear interface of the first Net-Net SBC.
2. Insert the other end of the cable into port 1 or port 2 on the rear interface of the second Net-Net SBC. We recommend that you use corresponding ports on the two systems. That is, use port 1 on both systems or use port 2 on both systems.
3. Perform software configuration for these interfaces as described in this chapter.

To cable Net-Net SBCs using multiple rear interface support:

1. Using a 10/100 category 5 crossover cable, insert one end into port 1 on the rear interface of the first Net-Net SBC.
2. Insert the other end of that cable into port 1 on the rear interface of the second Net-Net SBC to complete the first physical connection.
3. Using a second 10/100 category 5 cable, insert one end into port 2 on the rear interface of the first Net-Net SBC.
4. Insert the other end of this second cable in port 2 on the rear interface of the second Net-Net SBC to complete the second physical connection.
5. Perform software configuration for these interfaces as described in this chapter.

Virtual MAC Addresses

In order to create the HA node, you need to create virtual MAC addresses for the media interfaces. You enter these addresses in virtual MAC address parameters for physical interface configurations where the operation type for the interface is media.

The HA node uses shared virtual MAC (media access control) and virtual IP addresses for the media interfaces. When there is a switchover, the standby Net-Net SBC sends out an ARP message using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. Virtual MAC addresses are actually unused MAC addresses that based on the Net-Net SBC's root MAC address.

The MAC address is a hardware address that uniquely identifies each Net-Net SBC. Given that, the virtual MAC address you configure allows the HA node to appear as a single system from the perspective of other network devices. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted through the standby Net-Net SBC.

Depending on the type of physical layer cards you have installed, you can create MAC addresses as follows:

- One Ethernet (MAC) address for each configured one-port GigE physical interface card.
- Up to two Ethernet (MAC) addresses for each configured two-port GigE physical interface card.
- Up to four Ethernet (MAC) addresses for each configured 10/100 physical layer card.

ACLI Instructions and Examples

To create a virtual MAC address:

1. Determine the Ethernet address of the Net-Net SBC by using the ACCLI **show interfaces** command. This command only works if you have already set up physical interface configurations. Otherwise, you will get no output.

The example below shows you where the Ethernet address information appears; this sample has been shortened for the sake of brevity. For each type of physical interface card, the Net-Net SBC displays the following:

```
ACMEPACKET# show interfaces
f00 (media slot 0, port 0)
    Flags: UP BROADCAST MULTI CAST ARP RUNNING
    Type: GI GABI T_ETHERNET
    Admin State: enabled
    Auto Negotiation: enabled
    Internet address: 10.10.0.10      Vlan: 0
    Broadcast Address: 10.10.255.255
    Netmask: 0xfffff0000
    Gateway: 10.10.0.1
    Ethernet address is 00:08:25:01:07:64
```

2. Identify the root portion of the Ethernet (MAC) address.

Each Net-Net SBC has MAC addresses assigned to it according to the following format: 00:08:25:XX:YY:ZN where:

- 00:08:25 refers to Acme Packet
- XX:YY:ZN refers to the specific Net-Net SBC
- N is a 0-f hexadecimal value available for the Net-Net SBC

In this example, the root part of this address is 00:08:25:XX:YY:Z.

3. To create an unused MAC address (that you will use as the virtual MAC address) take the root MAC address you have just identified. Replace this N value with unused hexadecimal values for the Net-Net SBC: 8, 9, e, or f.

In other words, you change the last digit of the MAC address to either 8, 9, e, or f depending on which of those address are not being used.

For example, for an HA node with MAC address bases of 00:08:25:00:00:00 and 00:08:25:00:00:10, the following addresses would be available for use at virtual MAC addresses:

- 00:08:25:00:00:08
- 00:08:25:00:00:09
- 00:08:25:00:00:0e
- 00:08:25:00:00:0f
- 00:08:25:00:00:18
- 00:08:25:00:00:19
- 00:08:25:00:00:1e
- 00:08:25:00:00:1f

Corresponding media interfaces in HA nodes must have the same virtual MAC addresses. Given that you have various physical interface card options, the following points illustrate how virtual MAC address can be shared:

- If you are using one-port GigE physical interface cards, both the active Net-Net SBC and the standby Net-Net SBC might have the following virtual MAC address scheme for the slots:
 1. Slot 0—00:08:25:00:00:0e
 2. Slot 1—00:08:25:00:00:0f
- If you are using two-port GigE physical interface cards, both the active Net-Net SBC and the standby Net-Net SBC might have the following virtual MAC address scheme for the slots:
 1. Slot 0—00:08:25:00:00:0e and 00:08:25:00:00:0f
 2. Slot 1—00:08:25:00:00:1e and 00:08:25:00:00:1f
- If you are using 10/100 physical layer cards and you are using all eight of the ports, both the active Net-Net SBC and the standby Net-Net SBC might have the following virtual MAC address scheme for the slots:
 1. Slot 0—00:08:25:00:00:08, 00:08:25:00:00:09, 00:08:25:00:00:0e, and 00:08:25:00:00:0f
 2. Slot 1—00:08:25:00:00:18, 00:08:25:00:00:19, 00:08:25:00:00:1e and 00:08:25:00:00:1f
- 4. Note the virtual MAC addresses you have created so that you can reference them easily when you are configuring the physical interfaces for HA.

Configuring HA Node Connections

You can begin software configuration for your HA node after you have:

- Completed the steps for physical set-up and connection.
- Noted the target name of the Net-Net SBCs that make up the HA node.
- Configured the virtual MAC addresses that you need, according to the type of physical interface cards installed on your Net-Net SBC.

ACLI Instructions and Examples

If you are using HA, you need to set the physical interface configuration parameters described in this section to establish successful connections. These parameters are for rear and media interfaces.

Unless otherwise noted, all physical interface parameters should be configured as specified in the “Configuring Physical Interfaces” section of this guide.

To access physical interface menu in the ACLI:

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# **configure terminal**
2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(system)# **system**
3. Type **phy-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(system)# **phy-interface**
ACMEPACKET(phy-interface)#[/b]

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

Rear Interfaces

You can use port 1 (wancom1) or port 2 (wancom2) as interfaces to support HA. Do not use port 0 (wancom 0) as that port is reserved for carrying management traffic.

Make sure that the physical connections you have made on the rear panel of your Net-Net SBCs correspond to the configurations you enter for physical interfaces. You can connect Net-Net SBCs through multiple rear interfaces. For multiple rear interface connectivity, cable both port 1 and port 2 (wancom1 and wancom2) on one Net-Net SBC to port1 and port 2 on the other Net-Net SBC in the HA node.

The Net-Net SBC’s HA function depends heavily on health scores to determine the active and standby roles in an HA node. You can set the amount that will be subtracted from a Net-Net SBC’s health score in the event that a management interface fails for any reason. For example, a connection might become invalid or a cable might be removed inadvertently.

The following example shows how a configured physical interface will appear in the ACLI for an HA node:

phy-interface	
name	wancom1
operation-type	Maintenance
port	1
slot	0
virtual-mac	
wancom-health-score	20

To establish rear interfaces for use in an HA node using the ACCLI:

1. Access the physical interface menu.
2. **name**—Set a name for the interface using any combination of characters entered without spaces. For example: wancom1.
3. **operation-type**—Set this parameter to **maintenance**.
4. **slot**—Set this parameter to 0.
5. **port**—Set this parameter to 1 or 2.
6. **wancom-health-score**—Enter the number value between 0 and 100. This value will be subtracted from the Net-Net SBC's health score in the event that a rear interface link fails. We recommend that you change this value from its default (50), and set it to **20**.

This value you set here is compared to the active and emergency health score thresholds you establish in the Net-Net SBC HA node (redundancy) configuration.

7. For multiple rear interface support, configure the remaining, unused rear interfaces with the appropriate values.

The following example shows configuration for multiple rear interface support.

```
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# name wancom1
ACMEPACKET(phy-interface)# operation-type maintenance
ACMEPACKET(phy-interface)# port 1
ACMEPACKET(phy-interface)# wancom-heal th-score 20
ACMEPACKET(phy-interface)# done
ACMEPACKET(phy-interface)# name wancom2
ACMEPACKET(phy-interface)# operation-type maintenance
ACMEPACKET(phy-interface)# port 2
ACMEPACKET(phy-interface)# wancom-heal th-score 20
ACMEPACKET(phy-interface)# done
```

Media Interface Virtual MAC Addresses

To configure HA for the media interfaces in an HA node, you must set one or more virtual MAC addresses, according to the type of physical layer cards you have installed on your Net-Net SBC.

To set a virtual MAC address using the ACCLI:

1. Access the physical interface configuration.
2. Configure all relevant parameters as noted in the “Physical Interfaces” section of this guide’s *System Configuration* chapter.
Since virtual MAC addresses are used for media interfaces only, verify that the operation type is set to **media**.
3. **virtual-mac**—Enter the virtual MAC address that you have created using the steps in the [Virtual MAC Addresses \(879\)](#) section of this chapter.

Configuring HA Node Parameters

To establish a pair of Net-Net SBCs as an HA node, you need to configure basic parameters that govern how the Net-Net SBCs:

- Transition on switchover
- Share media and call state information

- Checkpoint configuration data

The following example shows what an HA configuration might look like in the ACCLI.

```
redundancy-config
  state          enabled
  log-level      WARNING
  heal-threshold 75
  emergency-threshold 50
  port           9090
  advertisement-time 500
  percent-drip   210
  initial-time   1250
  becoming-standby-time 45000
  becoming-active-time 100
```

You need to configure the two Net-Net SBCs to be HA node peers. To enable configuration checkpointing, you must to configure two peers in the ACCLI, one for the primary and one for the secondary Net-Net SBC. The HA node peers configuration also allows you to configure destinations for where to send health and state information. Unless you create Net-Net SBC peers and destinations configurations, HA will not work properly.

The following example shows what an HA configuration might look like in the ACCLI.

```
peer
  name          netnetsd1
  state         enabled
  type          Primary
  destination
    address     10.0.0.1:9090
    network-interface wancom1:0
  peer
  name          netnetsd2
  state         enabled
  type          Secondary
  destination
    address     10.0.0.2:9090
    network-interface wancom1:0
```

ACLI Instructions and Examples

To configure general HA node parameters using the ACCLI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure terminal
2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# system
3. Type **redundancy** and press <Enter>.
ACMEPACKET(system)# redundancy
From here, you configure basic HA node parameters. To view all basic HA node parameters, enter a ? at the system prompt.
4. **state**—Leave this parameter set to **enabled** for HA to work. To stop HA operation, set this parameter to **disabled**. The default value is **enabled**. The valid values are:

- enabled | disabled
5. **log-level**—Set the log level you want to use for the HA system process. The value you set in this field overrides any log level value you set for the entire Net-Net SBC in the system configuration process log level parameter. The default value is **INFO** which allows you to receive a moderate amount of detail. The valid values are:
 - emergency | critical | major | minor | warning | notice | info | trace | debug | detail
 6. **health-threshold**—Enter a value between 0 and 100 to set the health score at which the Net-Net SBCs in the HA node gracefully exchange active-standby roles. The default value is **75**. The valid range is:
 - Minimum—1
 - Maximum—100

For example, if this field is set to **75** and the active Net-Net SBC's health score falls below that point, the standby Net-Net SBC will take over the active role. However, Net-Net SBC will only take over the active role if its own health score is 75 or better.
 7. **emergency-threshold**—Enter the health score for the standby Net-Net SBC to become active immediately. The default value is **50**. The valid range is:
 - Minimum—0
 - Maximum—100

If the standby Net-Net SBC is initializing and the active Net-Net SBC's health score is below the health threshold, the standby Net-Net SBC will take the active role and there will be a graceful switchover. If the active Net-Net SBC's health score is below the emergency threshold, then the switchover will be immediate.

If the standby Net-Net SBC has a health score below the emergency threshold and the active Net-Net SBC is unhealthy, the active Net-Net SBC will not give up its active role.
 8. **advertisement-time**—Enter the number of milliseconds to set how often Net-Net SBCs in an HA node inform each other of their health scores.
- We recommend you leave this parameter set to its default, **500**. The valid range is:
- Minimum—50
 - Maximum—999999999
9. **percent-drift**—Enter the percentage of the advertisement time that you want one member of the HA node to wait before considering the other member to be out of service. For the standby Net-Net SBC, this is the time it will wait before taking the active role in the HA node. The default value is **210**. The valid range is:
 - Minimum—100
 - Maximum—65535
 10. **initial-time**—Enter the number of milliseconds to set the longest amount of time the Net-Net SBC will wait at boot time to change its state from initial to either becoming active or becoming standby. The default value is **1250**. The valid range is:
 - Minimum—5
 - Maximum—999999999

11. **becoming-standby-time**—Enter the number of milliseconds the Net-Net SBC waits before becoming standby, allowing time for synchronization. If it is not fully synchronized within this time, it will be declared out of service.

We recommend that you do not set this parameter below **45000**. If a large configuration is being processed, we recommend setting this parameter to **180000** to allow enough time for configuration checkpointing. The default value is **45000**. The valid range is:

- Minimum—5
- Maximum—999999999

12. **becoming-active-time**—Enter the number of milliseconds that the standby Net-Net SBC takes to become active in the event that the active Net-Net SBC fails or has an intolerably decreased health score. The default value is **100**. The valid range is:

- Minimum—5
- Maximum—999999999

To configure a Net-Net SBC as an HA node peer:

1. From the redundancy menu, type **peers** and press <Enter>.

```
ACMEPACKET(system)# redundancy
ACMEPACKET(redundancy)# peers
```

2. **state**—Enable or disable HA for this Net-Net SBC. The default value is **enabled**. The valid values are:

- enabled | disabled

3. **name**—Set the name of the HA node peer as it appears in the target name boot parameter.

This is also the name of your system that appears in the system prompt. For example, in the system prompt ACMEPACKET#, ACMEPACKET is the target name for that Net-Net SBC.

4. **type**—These values refer to the primary and secondary utility addresses in the network interface configuration. To determine what utility address to use for configuration checkpointing, set the type of Net-Net SBC: primary or secondary.

Note: You must change this field from **unknown**, its default. The valid values are:

- **primary**—Set this type if you want the Net-Net SBC to use the primary utility address.
- **secondary**—Set this type if you want the Net-Net SBC to use the secondary utility address.
- **unknown**—If you leave this parameter set to this default value, configuration checkpointing will not work.

To configure where to send health and state information within an HA node:

1. From the peers configuration, type destinations and press <Enter>.

```
ACMEPACKET(rdnyc-peer)# destinations
ACMEPACKET(rdnyc-peer-dest)#+
```

2. **address**—Set the destination IPv4 address and port of the other Net-Net SBC in the HA node to which this Net-Net SBC will send HA-related messages. This

value is an IPv4 address and port combination that you enter as: IPAddress:Port. For example, 10.0.0.1:9090.

- The IPv4 address portion of this value is the same as the IPv4 address parameter set in a network interface configuration of the other Net-Net SBC in the HA node.
 - The port portion of this value is the port you set in the Net-Net SBC HA Node/redundancy configuration for the other Net-Net SBC in the node.
3. **network-interface**—Set the name and subport for the network interface where the Net-Net SBC receives HA-related messages. Valid names are wancom1 and wancom2. This name and subport combination must be entered as name:subport; for example, **wancom1:0**.

The network interface specified in this parameter must be linked to a physical interface configured with rear interface parameters. The physical interface's operation type must be control or maintenance, and so the subport ID portion of this parameter is 0. The subport ID is the VLAN tag.

Synchronizing Configurations

You can synchronize the Net-Net SBCs in your HA node in the following ways:

- Automatically by setting up configuration checkpointing within the HA node
- Manually checking whether or not configurations in the HA node are synchronized, and then copying configuration data from one Net-Net SBC to the other in the node

When you initially configure a new HA node, you need to copy the configuration data manually from one Net-Net SBC to the other. However, once you have completed that process, you can configure your HA node to automatically synchronize configurations.

We recommend that you configure your HA node for configuration checkpointing because it is the most reliable way to ensure that both systems have the same configuration.

ACLI Instructions and Examples

To synchronize the systems in an HA node for the first time using the ACCLI:

1. Create a complete configuration on the active Net-Net SBC. This configuration should include all HA node parameters, including all rear interface configurations. Ensure the rear interfaces are configured so that information is sent and received across the HA node.
2. On the active Net-Net SBC, save the configuration you have created.
3. Reboot the active Net-Net SBC so that it will run using the configuration you have entered and saved.

Use the ACCLI **show health** command to see that the active Net-Net has come up without a peer. This changes after you copy the configuration to the standby Net-Net SBC and activate it.

4. On the standby Net-Net SBC, carry out the ACCLI **acquire-config** command to copy the configuration from the active Net-Net SBC. You use the **acquire-config** command with the IPv4 address of wancom 0 on the active Net-Net SBC.

```
ACMEPACKET2# acquire-config 192.168.12.4
```

The IPv4 address of wancom 0 on the active Net-Net SBC is the IPv4 address portion of the value you see displayed for the `i net on ethernet` boot parameter. When you view the boot parameters, the `i net on ethernet` value will look like this:

```
i net on ethernet (e) : 192.168.12.4:ffff0000
```

5. When the copying process (**acquire-config**) is complete, reboot the standby Net-Net SBC to activate the configuration. The booting process will begin, and start-up information will be displayed.
6. Confirm that the HA node now has synchronized configurations using the ACLI **display-current-cfg-version** and **display-running-cfg-version** commands:

```
ACMEPACKET1# display-current-cfg-version
Current configuration version is 3
ACMEPACKET1# display-running-cfg-version
Running configuration version is 3
ACMEPACKET2# display-current-cfg-version
Current configuration version is 3
ACMEPACKET2# display-running-cfg-version
Running configuration version is 3
```

As this example shows, all configuration versions—current and running—should have the same number. You can see that all configuration versions in this example are 3.

Using Configuration Checkpointing

The Net-Net SBC's primary and secondary utility addresses support configuration checkpointing, allowing the standby Net-Net SBC to learn configuration changes from the active Net-Net SBC. This means that you only have to enter configuration changes on the active Net-Net SBC for the configurations across the HA node to be updated.

Configuration checkpointing uses parameters in the network interface and in the SD HA Nodes/redundancy configurations.

If you are using configuration checkpointing, you also need to set up two Net-Net SBC peer configurations: one the primary, and one for the secondary.

ACLI Instructions and Examples

You need to first set applicable network interface configuration parameters, and then establish applicable parameters in the Net-Net SBC HA node (redundancy) configuration.

We recommend that you do not change the configuration checkpointing parameters in the redundancy configuration. Using the defaults, this feature will function as designed.

Note: Remember to set the appropriate **type** parameter in the HA node redundancy peers configuration. For more information about configuring peers, see page [886](#) of this chapter.

For the network interface, these parameters appear as they do in the following example when you use the ACLI. This example has been shortened for the sake of brevity.

<code>pri-utility-addr</code>	10.0.0.1
<code>sec-utility-addr</code>	10.0.0.2

For the Net-Net SBC HA node (redundancy) configuration, these parameters appear as they do in the following example when you use the ACCLI. This example has been shortened for the sake of brevity. You should not change these values without consultation from Acme Packet technical support or your Acme Packet Systems Engineer.

cfg-port	1987
cfg-max-trans	10000
cfg-sync-start-time	5000
cfg-sync-comp-time	1000

To configure HA configuration checkpointing in the ACCLI:

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

```
ACMEPACKET# configure terminal
```

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

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

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

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

```
ACMEPACKET(network-interface) #
```

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

4. **pri-utility-addr**—Enter the utility IPv4 address for the primary HA peer in an HA architecture.

This address can be any unused IPv4 address within the subnet defined for the network interface. For example, given a network interface of with the IPv4 address 168.0.4.15/24 (identifying the host associated with the network interface), the possible range of unused IPv4 addresses is 168.0.4.1 to 168.0.4.254. Your network administrator will know which IPv4 addresses are available for use.

5. **sec-utility-addr**—Enter the utility IPv4 address for the secondary Net-Net SBC peer in an HA architecture.

Usually, this IPv4 address is usually the next in the sequence up from the primary utility address. It is also generated from the range of unused IPv4 addresses within the subnet defined for the network interface.

6. Save your work and exit the network interface configuration.

```
ACMEPACKET(network-interface) # done
```

```
ACMEPACKET(network-interface) # exit
```

```
ACMEPACKET(system) #
```

7. Access the Net-Net SBC HA node/redundancy configuration by typing **redundancy** at the system prompt and then press <Enter>.

```
ACMEPACKET(system) # redundancy
```

```
ACMEPACKET(redundancy) #
```

Note: We strongly recommend that you keep the default settings for the parameters Steps 8 through 11.

8. **cfg-port**—Enter the port number for sending and receiving configuration checkpointing messages. Setting this to zero (0) disables configuration checkpointing. The default value is 1987. The valid values are:

- Minimum—0, 1025
 - Maximum—65535
9. **cfg-max-trans**—Enter the number of HA configuration checkpointing transactions that you want to store. The active Net-Net SBC maintains the transaction list, which is acquired by the standby Net-Net SBC. Then the standby system uses the list to synchronize its configuration with active system. The default value is **10000**. The valid range is:
- Minimum—0
 - Maximum— $2^{32}-1$
- Transactions include: modifications, additions, and deletions. If the maximum number of stored transactions is reached, the oldest transactions will be deleted as new transactions are added.
10. **cfg-sync-start-time**—Enter the number of milliseconds before the Net-Net SBC tries to synchronize by using configuration checkpointing. On the active Net-Net SBC, this timer is continually reset as the Net-Net SBC checks to see that it is still in the active role. If it becomes standby, it waits this amount of time before it tries to synchronize.
- We recommend you leave this field at its default value, **5000**, so that configuration checkpointing can function correctly. The valid range is:
- Minimum—0
 - Maximum— $2^{32}-1$
11. **cfg-sync-comp-time**—Enter the number of milliseconds that the standby Net-Net SBC waits before checkpointing to obtain configuration transaction information after the initial checkpointing process is complete.
- We recommend you leave this field at its default value, **1000**, so that configuration checkpointing can function correctly. The valid range is:
- Minimum—0
 - Maximum— $2^{32}-1$
12. Save your work and exit the redundancy configuration.

```
ACMEPACKET(redundancy)# done
ACMEPACKET(redundancy)# exit t
ACMEPACKET(system)#+
```

Manually Checking Configuration Synchronization

ACLI Instructions and Examples

To confirm that the systems in the HA node have synchronized configurations:

1. On the active Net-Net SBC in the Superuser menu, enter the following ALCI commands and press <Enter>. Note the configuration version numbers for comparison with those on the standby Net-Net SBC.
 - **display-current-cfg-version**—Shows the version number of the configuration you are currently viewing (for editing, updating, etc.).

```
ACMEPACKET1# display-current-cfg-version
Current configuration version is 30
```

- **display-running-cfg-version**—Shows the version number of the active configuration running on the Net-Net SBC.

```
ACMEPACKET1# di spl ay-runni ng-cfg-versi on
Running configuration version is 30
```

2. On the standby Net-Net SBC, enter the following ALCI commands and press <Enter>. Note the configuration version numbers for comparison with those on the active Net-Net SBC.

```
ACMEPACKET2# di spl ay-current-cfg-versi on
Current configuration version is 30
ACMEPACKET2# di spl ay-runni ng-cfg-versi on
Running configuration version is 30
```

3. Compare the configuration numbers. If the version numbers on the active Net-Net SBC match those on the standby Net-Net SBC, then the systems are synchronized.

If the version numbers do not match, you need to synchronize the Net-Net SBCs. You can do so using the ALCI **acquire-config** command as described in this chapter's [Synchronizing Configurations \(887\)](#).

Configuring Media Interface Link Detection and Gateway Polling

You can use media interface link detection and gateway polling globally on the Net-Net SBC, or you can override those global parameters on a per-network-interface basis.

- Use the Net-Net SBC HA node (redundancy) configuration to establish global parameters. When configured globally, they will appear like this in the ALCI:

gateway-heartbeat-interval	0
gateway-heartbeat-retry	0
gateway-heartbeat-timeout	1
gateway-heartbeat-heal th	0

- Use the network interface's gateway heartbeat configuration to override global parameters on a per-network-interface basis. When configured for the network interface, these parameters will appear like this in the ALCI:

gw-heartbeat		
state		enabl ed
heartbeat	0	
retry-count		0
retry-timeout		1
heal th-score		0

ALCI Instructions and Examples

To configure global media interface link detection and gateway polling:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# configure terminal
2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# system
3. Type **redundancy** and press <Enter>.
ACMEPACKET(system)# redundancy

From here, you can configure gateway heartbeat parameters. To view all gateway heartbeat parameters, enter a ? at the system prompt.

4. **gateway-heartbeat-interval**—Enter the number of seconds between heartbeats for the media interface gateway. Heartbeats are sent at this interval as long as the media interface is viable. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—65535
 5. **gateway-heartbeat-retry**—Enter the number of heartbeat retries (subsequent ARP requests) to send to the media interface gateway before it is considered unreachable. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—65535
 6. **gateway-heartbeat-timeout**—Enter the heartbeat retry time-out value in seconds. The default value is 1. The valid range is:
 - Minimum—0
 - Maximum—65535
- This parameter sets the amount of time between Net-Net SBC ARP requests to establish media interface gateway communication after a media interface gateway failure.
7. **gateway-heartbeat-health**—Enter the amount to subtract from the Net-Net SBC's health score if a media interface gateway heartbeat fails. If the value you set in the gateway time-out retry field is exceeded, this amount will be subtracted from the system's overall health score. The default value is 0. The valid range is:
 - Minimum—0
 - Maximum—100

To configure media interface link detection and gateway polling on a per-network-interface basis in the ACLI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
 2. Type **system** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# system
 3. Type **network-interface** and press <Enter>.
ACMEPACKET(system)# network-interface
 4. Type **gw-heartbeat** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(network-interface)# gw-heartbeat
ACMEPACKET(gw-heartbeat)#
- From here, you can configure gateway heartbeat parameters for the network interface. To view all gateway heartbeat parameters, enter a ? at the system prompt.
5. **state**—Enable or disable the gateway heartbeat feature. The default value is **enabled**. The valid values are:
 - enabled | disabled

6. **heartbeat**—Enter the number of seconds between heartbeats for the media interface gateway. Heartbeats are sent at this interval as long as the media interface is viable. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—65535

The value you configure in this field overrides any globally applicable value set in the gateway heartbeat interval parameter in the Net-Net SBC HA node (redundancy) configuration.
7. **retry-count**—Enter the number of heartbeat retries that you want sent to the media interface gateway before it is considered unreachable. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—65535
8. **retry-timeout**—Enter the heartbeat retry time-out value in seconds. The default value is 1. The valid range is:
 - Minimum—1
 - Maximum—65535

This parameter sets the amount of time between Net-Net SBC ARP requests to establish media interface gateway communication after a media interface gateway failure.
9. **health-score**—Enter the amount to subtract from the Net-Net SBC's health score if a media interface gateway heartbeat fails; this parameter defaults to 0. If the value you set in the retry-time-out field is exceeded, this amount will be subtracted from the system's overall health score. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—100

Signaling Checkpointing

You can configure your HA node to checkpoint signaling for SIP and MGCP.

SIP Signaling Checkpointing

In the SIP configuration, you can set parameters that enable SIP signaling checkpointing across an HA node.

When configured, these parameters will appear in the ACLI as they do in example below.

Note: This example shows the default values being used, and we recommend that you do not change these values from their defaults.

red-sip-port	1988
red-max-trans	10000
red-sync-start-time	5000
red-sync-comp-time	1000

ACLI Instructions and Examples

To configure SIP signaling checkpointing across an HA node in the ACLI:

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

ACMEPACKET# **config terminal**

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

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

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

ACMEPACKET(session-router)# **sip-config**

From here, you can configure SIP parameters for HA nodes. To view all SIP configuration parameters, enter a ? at the system prompt.

When configuring SIP for HA, you only need to set the parameters addressed in this procedure.

4. **red-sip-port**—Enter the port on which SIP signaling checkpointing messages are sent and received. The default value is **1988**. A value of **0** disables the SIP signaling checkpointing. The valid range is:

- Minimum—0, 1024
- Maximum—65535

5. **red-max-trans**—Enter the maximum size of the transaction list, or how many SIP transactions you want to store in memory at one time. Oldest transactions will be discarded first in the event that the limit is reached. The default value is **10000**. The valid range is:

- Minimum—0
- Maximum—999999999

6. **red-sync-start-time**—Enter the number of milliseconds before the Net-Net SBC will try to synchronize its signaling state checkpointing.

If the active Net-Net SBC is still adequately healthy, this timer will simply reset itself. If for any reason the active Net-Net SBC has become the standby, it will start to checkpoint with the newly active system when this timer expires.

We recommend that you leave this parameter set to its default, **5000**. The valid range is:

- Minimum—0
- Maximum—999999999

7. **red-sync-comp-time**—Enter the number of milliseconds representing how frequently the standby Net-Net SBC checkpointing with the active Net-Net SBC to obtain the latest SIP signaling information. The first interval occurs after initial synchronizations of the systems.

We recommend that you leave this parameter set to its default, **1000**. The valid range is:

- Minimum—0
- Maximum—999999999

MGCP Configuration for HA Nodes

In the MGCP configuration, you can set parameters that enable MGCP signaling checkpointing across an HA node.

Note: When an HA node is configured to use hosted NAT traversal (HNT) for MGCP, you need to set the audit interval parameter in the MGCP configuration to one-third of the network connectivity time-out

for your NAT device. AUEP messages will be sent to keep the pinhole open in the NAT device; they will continue to be sent if a switchover occurs. Refer to the MGCP Configuration chapter of this guide for information about how to set this parameter.

When configured, these parameters will in the ACI as follows:

red-mgcp-port	1986
red-max-trans	10000
red-sync-start-time	5000
red-sync-comp-time	1000

MGCP Media Session Replication

With Net-Net 4000 Release C5.0, the Net-Net SBC's support for media session replication between active and standby systems in an HA node has been improved to support binary encoding of replicated data. Now, not only are MGCP connections statistics are more accurately reported between active and standby, but MGCP session and signaling is more reliably and efficiently duplicated between active and standby.

Note that when upgrading from another release to Net-Net 4000 Release C5.0, the ASCII format is still used for the purpose of backward compatibility. Only when both Net-Net 4000 systems in an HA node are running Net-Net 4000 Release C5.0 will the binary format to support these improvements be used.

ACI Instructions and Examples

To configure MGCP signaling checkpointing across an HA node in the ACI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# **media-manager**
3. Type **media-manager-config** and press <Enter>.
ACMEPACKET(media-manager)# **media-manager-config**
4. **red-mgcp-port**—Enter the port on which MGCP signaling checkpointing messages are sent and received.
Setting this parameter to **0** disables MGCP signaling checkpointing.
The default value is **1986**. The valid range is:
 - Minimum—0, 1025
 - Maximum—65535
5. **red-max-trans**—Enter the maximum size of the transaction list, or how many MGCP transactions you want to store in memory at one time. Oldest transactions will be discarded first in the even that the limit is reached. The default value is **1000**. The valid range is:
 - Minimum—0
 - Maximum—999999999
6. **red-sync-start-time**—Enter the number of milliseconds that the active Net-Net SBC checks to confirm that it is still the active system in the HA node. If the active Net-Net SBC is still adequately healthy, this timer will simply reset itself. If for any reason the active Net-Net SBC has become the standby, it will start to checkpoint with the newly active system when this timer expires.

We recommend that you leave this parameter set to its default, **5000**. The valid range is:

- Minimum—0
- Maximum—999999999

7. **red-sync-comp-time**—Enter the number of milliseconds representing how frequently the standby Net-Net SBC checkpointing with the active Net-Net SBC to obtain the latest MGCP signaling information. The first interval occurs after initial synchronizations of the systems.

We recommend that you leave this parameter set to its default, **1000**. The valid range is:

- Minimum—0
- Maximum—999999999

Media State Checkpointing

By default, the Net-Net SBC performs media checkpointing across the HA node for all signaling protocols. You can keep the default port set for redundancy media flows.

H.323 media high availability is supported through a TCP socket keep-alive, which determines whether or not the other end of a TCP/IP network connection is still in fact connected. This type of checkpointing prevents the listening side of a connection from waiting indefinitely when a TCP connection is lost. When there is a switchover in the HA node, the system that has just become active takes over sending TCP keep-alives. Media continues to flow until the session ends or the flow guard timers expire.

This parameter will appear in the ACLI as follows:

red-flow-port	1985
---------------	------

ACLI Instructions and Examples

To configure media state checkpointing across an HA node in the ACLI:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# media-manager
3. Type **media-manager-config** and press <Enter>.
ACMEPACKET(media-manager)# media-manager-config
4. **red-flow-port**—Enter the port number for checkpointing media flows associated with the HA interface. This is the port where media flow checkpoint message are sent and received.

Setting this field to **0** disables media state checkpointing. The default value is **1985**. The valid range is:

- Minimum—0, 1025
- Maximum—65535

HA Media Interface Keepalive

In an HA node, it is possible for the two systems in the node to lose communication via the management (rear, wancom) interfaces. For example, wancom 1 and wancom 2 might become disconnected, and cause the heartbeat synchronization to fail. This type of failure causes communication errors because both systems try to assume the active role and thereby access resources reserved for the active system.

To avoid these types of conditions, you can enable an option instructing the standby system to take additional time before going to the active state. This check occurs through the system's media interfaces. Using it, the standby can determine whether or not there has been a true active failure.

In cases when the standby determines the active system has not truly failed, it will go out of service because it will have determined it no longer has up-to-date data from its active counterpart. You can restore functionality by re-establishing management (rear) interface communication between the system in the node, and then re-synchronizes the standby by rebooting it.

How It Works

When you enable the media interface keepalive, the standby system in the HA node sends ARP requests to determine if the media interfaces' virtual IP address are active. There are two possible outcomes:

- If it receives responses to its ARP requests, the standby takes itself out of service—to prevent a conflict with the active.
- If it does not receive responses to its ARP requests within a timeout value you set, then standby assumes the active role in the HA node.

Impact to Boot-Up Behavior

With the HA media interface keepalive enabled, the Net-Net SBC might be in the "initial" state longer than if the feature were disabled because it requires more information about the media (front) interfaces.

ACLI Instructions and Examples

You turn the HA media interface keepalive on by setting a timeout value for the standby to receive responses to its ARP requests before it assumes the active role in the HA node. Keeping this parameter set to 0, its default, disables the keepalive

To enable the HA media interface keepalive:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
ACMEPACKET(configure)#
2. Type **system** and press <Enter>.
ACMEPACKET(configure)# **system**
ACMEPACKET(system)#
3. Type **redundancy** and press <Enter>.
ACMEPACKET(session-router)# **redundancy**
ACMEPACKET(redundancy)#

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.

4. **media-if-peercheck-time**—Enter the amount of time in milliseconds for the standby system in an HA node to receive responses to its ARP requests via the media interface before it takes over the active role from its counterpart.
The default is 0, which turns the HA media interface keepalive off. The maximum value is 500 milliseconds.
5. Save and activate your configuration.

RTC Notes

Starting in Release 4.1, the HA configuration is supported for real-time configuration (RTC). However, not all of the HA-related parameters are covered by RTC because of the impact on service it would cause to reconfigure these parameters dynamically.

This section sets out what parameters you should not dynamically reconfigure, or should dynamically reconfigure with care.

HA

Changes to the following ACLI parameters will have the noted consequences when dynamically reconfigured:

- **cfg-max-trans**—Changing this value could cause the activation time to lengthen slightly
- **init-time, becoming-standby-time, and becoming-active-time**—Changes take place only if the system is not transitioning between these states; otherwise the system waits until the transition is complete to make changes
- **percent-drift and advertisement-time**—Changes are communicated between nodes in the HA pair as part of regular health advertisements

In addition, the following parameters are not part of the RTC enhancement, for the reason specified in the right-hand column.

Parameter	Impact
state	Disrupts service
port	Disrupts service; leaves Net-Net SBCs in an HA node without a means of communicating with each other
cfg-port	Disrupts service; leaves Net-Net SBCs in an HA node without a means of communicating with each other
cfg-max-trans	Disrupts service
cfg-sync-start-time	Disrupts configuration replication
cfg-sync-comp-time	Disrupts configuration replication

Protocol-Specific Parameters and RTC

In addition, you should not change any of the parameters related to HA that are part of protocol or media management configurations that are used for protocol/media checkpointing. These are:

- SIP configuration
 - **red-max-trans**
 - **red-sync-start-time**
 - **red-sync-comp-time**

- MGCP Configuration
 - **red-mgcp-port**
 - **red-max-trans**
 - **red-sync-start-time**
 - **red-sync-comp-time**
- Media Manager configuration
 - **red-flow-port**
 - **red-mgcp-port**
 - **red-max-trans**
 - **red-sync-start-time**
 - **red-sync-comp-time**

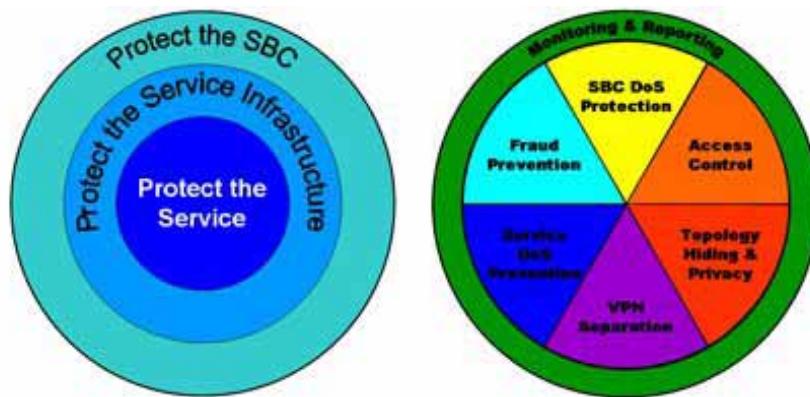
Introduction

This chapter explains Net-Net SBC 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 SBC). In addition, Net-Net SBC security lets legitimate users still place calls during attack conditions; protecting the service itself.

Security Overview

Net-Net SBC security includes the Net-SAFE framework's numerous features and architecture designs. Net-SAFE is a requirements framework for the components required to provide protection for the Session Border Controller (SBC), the service provider's infrastructure equipment (proxies, gateways, call agents, application servers, and so on), and the service itself.

The following diagrams illustrate Net-SAFE:



Each of Net-SAFE's seven functions consists of a collection of more specific features:

- Session border controller DoS protection: autonomic, SBC self-protection against malicious and non-malicious DoS attacks and overloads at Layers 2 to 4 (TCP, SYN, ICMP, fragments, and so on) and Layers 5 to 7 (SIP signaling floods, malformed messages, and so on).
- Access control: session-aware access control for signaling and media using static and dynamic permit/deny access control lists (ACLs) at layer 3 and 5.
- Topology hiding and privacy: complete infrastructure topology hiding at all protocol layers for confidentiality and attack prevention security. Also, modification, removal or insertion of call signaling application headers and fields. Includes support for the SIP Privacy RFC.

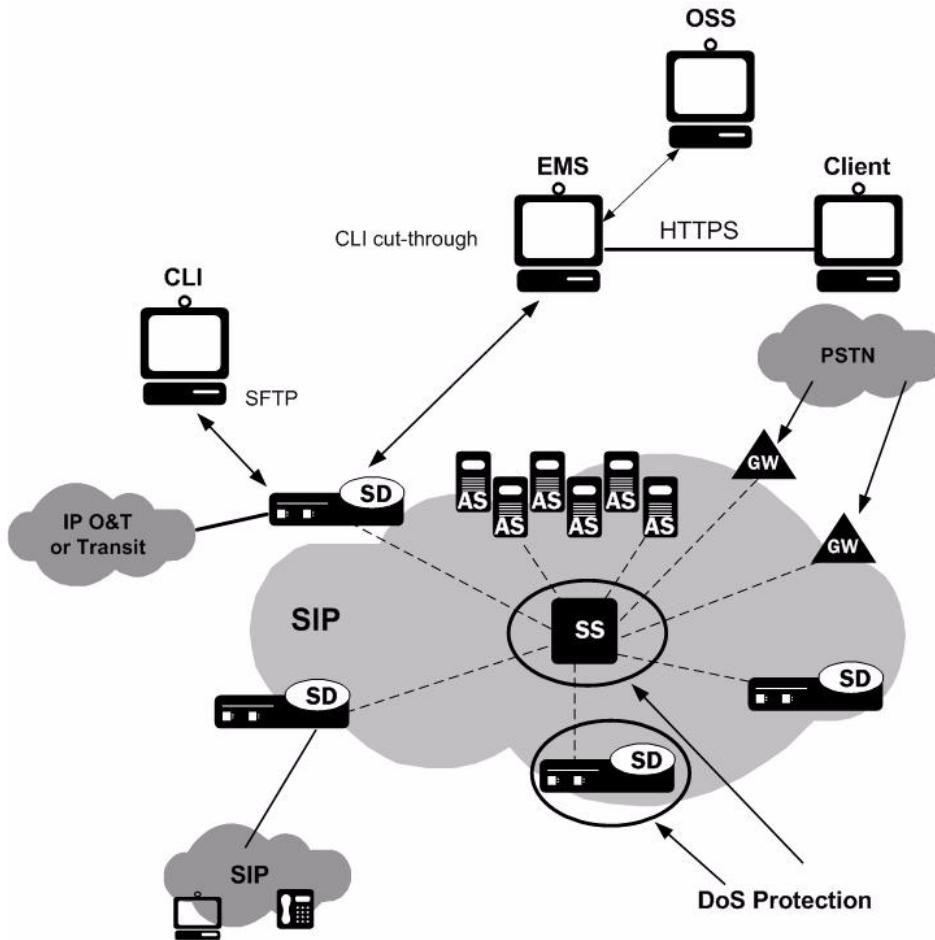
- VPN separation: support for Virtual Private Networks (VPNs) with full inter-VPN topology hiding and separation, ability to create separate signaling and media-only VPNs, and with optional intra-VPN media hair-pinning to monitor calls within a VPN.
- Service infrastructure DoS prevention: per-device signaling and media overload control, with deep packet inspection and call rate control to prevent DoS attacks from reaching service infrastructure such as SIP servers, softswitches, application servers, media servers or media gateways.
- Fraud prevention: session-based authentication, authorization, and contract enforcement for signaling and media; and service theft protection.
- Monitoring and reporting: audit trails, event logs, access violation logs and traps, management access command recording, Call Detail Records (CDRs) with media performance monitoring, raw packet capture ability and lawful intercept capability. The monitoring method itself is also secured, through the use of SSH and SFTP, and through the ability to use a separate physical Ethernet port for management access.

Denial of Service Protection

This section explains the Denial of Service (DoS) protection for the Net-Net SBC. The Net-Net SBC 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 SBC 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 SBC host processor from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Volume-based attack (flood) of valid or invalid call requests, signaling messages, and so on.
- Overload of valid or invalid call requests from legitimate, trusted sources

The following diagram illustrates DoS protection applied to the softswitch and to the Net-Net SBC.



Levels of DoS Protection

The multi-level Net-Net SBC DoS protection consists of the following strategies:

- Fast path filtering/access control: access control for signaling packets destined for the Net-Net SBC host processor as well as media (RTP) packets. The Net-Net SBC 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 SBC realm (although the actual devices can be dynamically trusted or denied by the Net-Net SBC based on configuration). You do not have to provision every endpoint/device on the Net-Net SBC, 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 SBC 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 SBC's normal call processing; and subsequently not overwhelm systems beyond it.

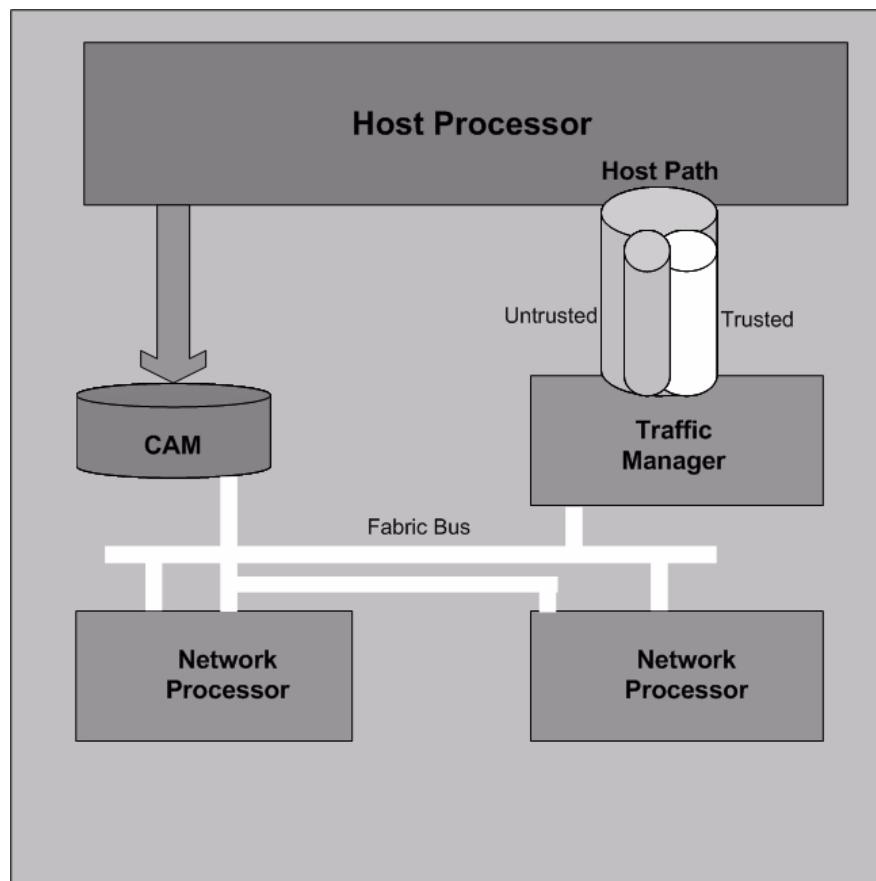
The Net-Net SBC must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent. At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Net-Net SBC maintains two host paths, one for each class of traffic (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 SBC's host path. The Net-Net SBC 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 SBC can support is 16K (on 32K CAM / IDT CAM).

The Traffic Manager has two pipes, trusted and untrusted, for the signaling path. Each signaling packet destined for the host CPU traverses one of these two pipes.



Trusted Path

Packets from trusted devices travel through the trusted pipe in their own individual queues. In the Trusted path, each trusted device flow has its own individual queue (or pipe). The Net-Net SBC can dynamically add device flows to the trusted list by promoting them from the Untrusted path based on behavior; or they can be statically provisioned.

Trusted traffic is put into its own queue and defined as a device flow based on the following:

- source IP address
- source UDP/TCP port number
- destination IP address
- destination UDP/TCP port (SIP or MGCP interface to which it is sending)
- realm it belongs to, which inherits the Ethernet interface and VLAN it came in on

For example, SIP packets coming from 10.1.2.3 with UDP port 1234 to the Net-Net SBC 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 SBC 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 SBC 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 SBC (therefore it is trusted, but not completely).

Address Resolution Protocol Flow

The Address Resolution Protocol (ARP) packets are given their own trusted flow with the bandwidth limitation of 8 Kbps. ARP packets are able to flow smoothly, even when a DoS attack is occurring.

Untrusted Path

Packets (fragmented and unfragmented) that are not part of the trusted or denied list travel through the untrusted pipe. In the untrusted path, traffic from each user/device goes into one of 2048 queues with other untrusted traffic. Packets from a single device flow always use the same queue of the 2048 untrusted queues, and 1/2048th of the untrusted population also uses that same queue. To prevent one untrusted endpoint from using all the pipe's bandwidth, the 2048 flows defined within the path are scheduled in a fair-access method. As soon as the Net-Net SBC decides the device flow is legitimate, it will promote it to its own trusted queue.

All 2048 untrusted queues have dynamic sizing ability, which allows one untrusted queue to grow in size, as long as other untrusted queues are not being used proportionally as much. This dynamic queue sizing allows one queue to use more than average when it is available. For example, in the case where one device flow represents a PBX or some other larger volume device. If the overall amount of untrusted packets grows too large, the queue sizes rebalance, so that a flood attack or DoS attack does not create excessive delay for other untrusted devices.

In the usual attack situations, the signaling processor detects the attack and dynamically demotes the device to denied in the hardware by adding it to the deny ACL list. Even if the Net-Net SBC does not detect an attack, the untrusted path gets serviced by the signaling processor in a fair access mechanism. An attack by an untrusted device will only impact 1/1000th of the overall population of untrusted devices, in the worst case. Even then there's a probability of users in the same 1/1000th percentile getting in and getting promoted to trusted.

IP Fragment Packet Flow

All fragment packets are sent through their own 1024 untrusted flows in the Traffic Manager. The first ten bits (LSB) of the source address are used to determine which fragment-flow the packet belongs to. These 1024 fragment flows share untrusted bandwidth with already existing untrusted-flows. In total, there are 2049 untrusted flows: 1024-non-fragment flows, 1024 fragment flows, and 1 control flow.

Fragmented ICMP packets are qualified as ICMP packets rather than fragment packets. Fragment and non-fragmented ICMP packets follow the trusted-ICMP-flow in the Traffic Manager, with a bandwidth limit of 8Kbs.

Fragment Packet Loss Prevention

You can set the maximum amount of bandwidth (in the **max-untrusted-signaling** parameter) you want to use for untrusted packets. However, because untrusted and fragment packets share the same amount of bandwidth for policing, any flood of untrusted packets can cause the 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.

Static and Dynamic ACL Entry Limits

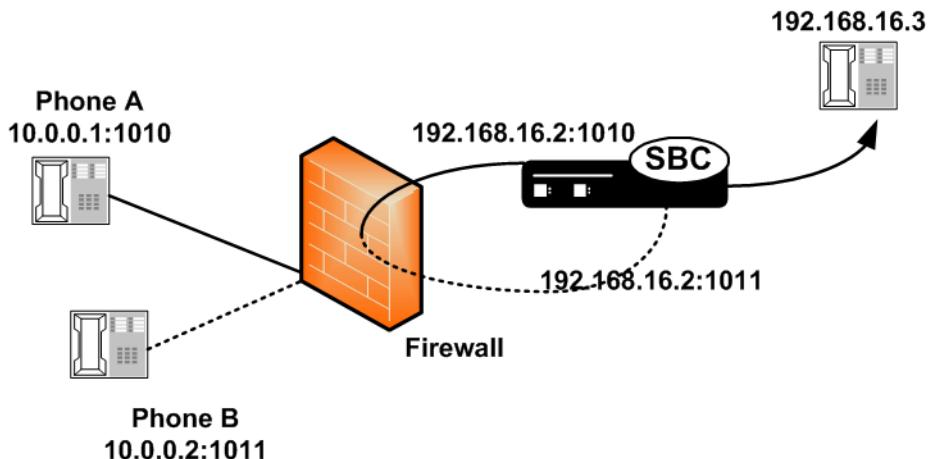
The Net-Net SBC can simultaneously police a maximum of 16,000 trusted device flows, while at the same time denying an additional 16,000 attackers. If all 16,000 trusted entries and 16,000 denied entries are being used, the Net-Net SBC can handle at most 32,000 simultaneous media flows (which is 16,000 simultaneous calls for normal voice calls). The usage is dynamic, so that when the Net-Net SBC uses fewer trusted or denied entries, additional capacity for call media handling is freed. These limits let the Net-Net SBC 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 SBC 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 SBC would also deny all other users behind the same NAT or firewall. This would be true even for endpoints behind the firewall that had not crossed threshold limits you set for their realm; all endpoints behind the firewall would go out of service. In the following diagram, both Phone A and Phone B would be denied because their IP addresses would be translated by the firewall to the same IPv4 address (192.168.16.2).

However, dynamic deny for HNT allows the Net-Net SBC to determine, based on the UDP/TCP port, which endpoints should be denied and which should be allowed. The Net-Net SBC can determine that even though multiple endpoints originating behind a firewall appear with the same IPv4 address, those addresses use different ports and are unique.

As shown in the diagram below, the ports from Phone A and Phone B remain unchanged. This way, if Phone A violates the thresholds you have configured, the Net-Net SBC can block traffic from Phone A while still accepting traffic from Phone B.



Host and Media Path Protection Process

The Net-Net SBC 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 SBC 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 SBC can dynamically promote and

demote device flows based on the behavior, and thus dynamically creates trusted, untrusted, and denied list entries.

Access Control for Hosts

ACLs are supported for all VoIP signaling protocols on the Net-Net SBC: SIP, H.323, and MGCP. The Net-Net SBC 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 SBC 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 SBC 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 SBC 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 signaling (SIP, H.323, MGCP) are allowed, which reduces the chance of RTP hijacking. Media access depends on both the destination and source RTP/RTCP UDP port numbers being correct, for both sides of the call.

Host Path Traffic Management

The host path traffic management consists of the dual host paths discussed earlier:

- Trusted path is for traffic classified by the Net-Net SBC as trusted. You can initially define trusted traffic by ACLs, as well as by dynamically promoting it through successful SIP or MGCP registration, or a successful call establishment. You can configure specific policing parameters per ACL, as well as define default policing values for dynamically-classified flows. Traffic for each trusted device flow is limited from exceeding the configured values in hardware. Even an attack from a trusted, or spoofed trusted, device cannot impact the system.
- Untrusted path is the default for all unknown traffic that has not been statically provisioned otherwise. For example, traffic from unregistered endpoints. Pre-configured bandwidth policing for all hosts in the untrusted path occurs on a per-queue and aggregate basis.

Traffic Promotion

Traffic is promoted from untrusted to trusted list when the following occurs:

- successful SIP registration for SIP endpoints
- successful RSIP response for MGCP endpoints
- successful session establishment for SIP or MGCP calls

Malicious Source Blocking

Malicious source blocking consists of monitoring the following metrics for each source:

- SIP transaction rate (messages per second)
- SIP call rate (call attempts per second)
- Nonconformance/invalid signaling packet rate

Device flows that exceed the configured invalid signaling threshold, or the configured valid signaling threshold, within the configured time period are demoted, either from trusted to untrusted, or from untrusted to denied classification.

Blocking Actions

Blocking actions include the following:

- Dynamic deny entry added, which can be viewed through the ACLI.
- SNMP trap generated, identifying the malicious source

Dynamically added deny entries expire and are promoted back to untrusted after a configured default deny period time. You can also manually clear a dynamically added entry from the denied list using the ACLI.

Protecting Against Session Agent Overloads

You can prevent session agent overloads with registrations by specifying the registrations per second that can be sent to a session agent.

ARP Flood Protection Enhancements

Enhancements have been made to the way the Net-Net SBC provides ARP flood protection. In releases prior to Release C5.0, 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.

High-Capacity CAM

The Net-Net 4250 SBC can be upgraded to use high-capacity, 256K CAM, a field upgradable hardware unit. The increased capacity offered by the 256K CAM expands the dynamic and static ACL capacity to enhance DoS protection.

There is no special configuration required for using the 256K CAM. However, using it changes the default values for DoS flow classification allocation, a feature added in Net-Net Release 4.1. This feature lets you set limits for three of the four types of flow classifications used for DoS functionality. The following table lists the default values for these parameters, which this system uses unless you explicitly set new ones:

Table 4:

ACLI parameter	64K CAM	256K CAM
min-media-allocation	32K CAM entries	32K CAM entries
min-trusted-allocation	4K CAM entries	60K CAM entries
deny-allocation	1K CAM entries	32K CAM entries

For each type of CAM, the remainder is available in a floating pool.

Minimum hardware requirements: board must be at rev 1.03 with a FPGA of rev 1.13.

Contact your Acme Packet sales representative if you are interested in purchasing the 256K CAM upgrade.

Dynamic Demotion for NAT Devices

In addition to the various ways the Net-Net SBC already allows you to promote and demote devices to protect itself and other network elements from DoS attacks, it can now block off an entire NAT device. The Net-Net SBC can detect when a configurable number of devices behind a NAT have been blocked off, and then shut off the entire NAT's access.

This dynamic demotion of NAT devices can be enabled for an access control (ACL) configuration or for a realm configuration. When you enable the feature, the Net-Net SBC tracks the number of endpoints behind a single NAT that have been labeled untrusted. It shuts off the NAT's access when the number reaches the limit you set.

The demoted NAT device then remains on the untrusted list for the length of the time you set in the **deny-period**.

Configuring DoS Security

This section explains how to configure the Net-Net SBC for DoS protection.

Configuration Overview

Configuring Net-Net SBC DoS protection includes masking source IP and port parameters to include more than one match and configuring guaranteed minimum bandwidth for trusted and untrusted signaling path. You can also configure signaling path policing parameters for individual source addresses. Policing parameters are defined as peak data rate (in bytes/sec), average data rate (in bytes/sec), and maximum burst size.

You can configure deny list rules based on the following:

- ingress realm
- source IP address
- source port
- transport protocol (TCP/UDP)

- application protocol (SIP, MGCP, H.323)

Changing the Default Net-Net SBC Behavior

The Net-Net SBC 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 signalling interfaces/port of the realm named access, are allowed. The packets go into untrusted queues until they are dynamically demoted or promoted based on their behavior. All other packets are denied/dropped.

- Configure a realm called access and set the trust level to low and the address prefix to 12.34.0.0/16.
- Configure a static ACL with a source prefix of 12.34.0.0/16 with the trust level set to low for the realm named access.

Example 2: Classifying the Packets as Trusted

Building on Example 1, this example shows how to classify all packets from 12.34.0.0/16 to the realm signalling interfaces as trusted and place them in a trusted queue. All other packets from outside the prefix range destined to the realm's signalling interfaces are allowed and classified as untrusted; then promoted or demoted based on behavior.

You do this by adding a global permit untrusted ACL (source 0.0.0.0) for each signalling 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 SBC to also add a permit trusted ACL with a source prefix of 12.34.0.0/16 for each signalling 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 SBC from installing the global permit (0.0.0.0) untrusted ACL.

- Configure a realm with a trust level of none.
- Configure static ACLs for that realm with the same source address prefix as the realm's address prefix, and set the trust level to any value.

The Net-Net SBC installs only the static ACLs you configure.

Configuring Access Control Lists

To configure access control:

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

```
ACMEPACKET# config terminal
```

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

```
ACMEPACKET(config)# 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. **source-address**—Enter the source IPv4 address and port number for the host in the following format:

```
<IP address>[/number of address bits] [:<port>] [/<port bits>]
```

For example:

10.0.0.1/24: 5000/14

10.0.0.1/16

10.0.0.1/24: 5000

10.0.0.1: 5000

You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. The default value is **0.0.0.0**.

6. **destination-address**—(Is ignored if you configure an application protocol in step 7.) Enter the destination IPv4 address and port for the destination in the following format:

```
<IP address>[/number of address bits] [:<port>] [/<port bits>]
```

You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. The default value is **0.0.0.0**.

7. **application-protocol**—Enter the application protocol type for this ACL entry. The valid values are:

- SIP | H.323 | MGCP

8. **transport-protocol**—Select the transport-layer protocol configured for this ACL entry. The default value is **ALL**. The valid values are:

- ALL | TCP | UDP

9. **access**—Enter the access control type or trusted list based on the trust-level parameter configuration for this host. The default value is **permit**. The valid values are:

- **permit**—Puts the entry into the untrusted list. The entry is promoted or demoted according to the trust level set for this host.

- **deny**—Puts the entry in the deny list.

10. **average-rate-limit**—Indicate the sustained rate in bytes per second for host path traffic from a trusted source within the realm. The default value is **0**. A value of **0** means policing is disabled. The valid range is:
 - Minimum—0
 - Maximum—999999999
11. **trust-level**—Indicate the trust level for the host with the realm. The default value is **none**. The valid values are:
 - **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - **low**—Host can be promoted to the trusted list or demoted to the deny list.
 - **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - **high**—Host is always trusted.
12. **invalid-signal-threshold**—Indicate the rate of signaling messages per second to be exceeded within the tolerance-window that causes a demotion event. This parameter is only valid when **trust-level** is configured as low or medium. A value of 0 means no threshold. The default value is **0**. The valid range is:
 - Minimum—Zero (0) is disabled.
 - Maximum—999999999

If the number of invalid messages exceeds this value within the tolerance window, the host is demoted.
13. **maximum-signal-threshold**—Set the maximum number of signaling messages the host can send within the tolerance window. The value you enter here is only valid when the trust level is low or medium. The default value is **0**, disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—999999999

If the number of messages received exceeds this value within the tolerance window, the host is demoted.
14. **untrusted-signal-threshold**—Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is **0**, disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—999999999
15. **deny-period**—Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is **30**. The valid range is:
 - Minimum—0
 - Maximum—999999999
16. **nat-trust-threshold**—Enter maximum acceptable number of untrusted endpoints allowed before the Net-Net SBC demotes the entire NAT device to untrusted (dynamic demotion of NAT devices. The default is 0, meaning dynamic demotion of NAT devices is disabled.

The following example shows access control configured for a host in the external realm.

```

access-control
    real-m-id          external
    source-address      192.168.200.215
    destination-address 192.168.10.2:5000
    application-protocol SIP
    transport-protocol ALL
    access              permit
    average-rate-limit 3343
    trust-level         low
    invalid-signal-threshold 5454
    maximum-signal-threshold 0
    untrusted-signal-threshold 0
    deny-period         0

```

The following example of how to configure a black-list entry:

```

access-control
    real-m-id          external
    source-address      192.168.200.200
    destination-address 192.168.10.2:5000
    application-protocol SIP
    transport-protocol ALL
    access              deny
    average-rate-limit 0
    trust-level         none
    invalid-signal-threshold 0
    maximum-signal-threshold 0
    untrusted-signal-threshold 0
    deny-period         0

```

Host Access Policing

You can configure the Net-Net SBC 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# config terminal
2. Type **media-manager** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(config)# media-manager
3. Type **media-manager** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
4. **max-signaling-bandwidth**—Set the maximum overall bandwidth available for the host path in bytes per second, which includes signaling messages from trusted and untrusted sources. It also includes any Telnet and FTP traffic on media ports. The default value is **1000000**. The valid range is:
 - Minimum—71000
 - Maximum—10000000
5. **max-untrusted-signaling**—Set the percentage of the maximum signaling bandwidth you want to make available for messages coming from untrusted

sources. This bandwidth is only available when not being used by trusted sources. The default value is **100**. The valid range is:

- Minimum—1
 - Maximum—100
6. **min-untrusted-signaling**—Set the percentage of the maximum signaling bandwidth you want reserved for the untrusted sources. The rest of the bandwidth is available for trusted resources, but can also be used for untrusted sources (see **max-untrusted-signaling**). The default value is **30**. The valid range is:
 - Minimum—1
 - Maximum—100
 7. **fragment-msg-bandwidth**—Enter the amount of bandwidth to use for the fragment packet queue. If you leave this parameter set to 0, then the Net-Net SBC will use the same queue for and share bandwidth between untrusted packets and fragment packets. The default value is zero (**0**). The valid range is:
 - Minimum—0
 - Maximum—10000000
 8. **tolerance-window**—Set the size of the window used to measure host access limits. The value entered here is used to measure the invalid message rate and maximum message rate for the realm configuration. The default value is **30**. The valid range is:
 - Minimum—0
 - Maximum—999999999

The following example shows a host access policing configuration.

medi a-manager	
state	enabl ed
l atching	enabl ed
f low-ti me-l imit	86400
i niti al -guard-timer	300
subsq-guard-timer	300
tcp-f low-ti me-l imit	86400
tcp-i niti al -guard-timer	300
tcp-subsq-guard-timer	300
tcp-number-of-ports-per-fl ow	2
hnt-rtcp	di sabl ed
al gd-log-l evel	WARNI NG
mbcd-log-l evel	WARNI NG
home-real m-id	
red-f low-port	1985
red-mgcp-port	1986
red-max-trans	10000
red-sync-start-ti me	5000
red-sync-comp-ti me	1000
max-signal i ng-bandwi dth	1000000
max-untrusted-signal i ng	50
mi n-untrusted-signal i ng	30
tol erance-wi ndow	30
rtcp-rate-l imit	0

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: 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 default value is **32000**. The valid range is:
 - Minimum—2000
 - Maximum—200000
5. Save your configuration.
6. 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(real-m-config)#
4. **addr-prefix**—Set the IP address prefix used to determine if an IP address is associated with the realm. This value is then associated with the ACLs you create to determine packet access. The default value is **0.0.0.0**.

5. **average-rate-limit**—Set the sustained rate for host path traffic from a trusted source within the realm in bytes per second. The default value is zero (0), disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—4294967295
6. **access-control-trust-level**—Set the trust level for the host within the realm. The default value is **none**. The valid values are:
 - **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - **low**—Host can be promoted to the trusted list or demoted to the deny list.
 - **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - **high**—Host is always trusted.
7. **invalid-signal-threshold**—Set the acceptable invalid signaling rate within the window of tolerance. If the Net-Net SBC 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 set to low or medium. The default value is zero (0), disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—4294967295
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 SBC exceeds the number set here. Valid only when the trust level is set to low or medium. The default value is zero (0), disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—4294967295
9. **untrusted-signal-threshold**—Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is zero (0), disabling the parameter. The valid range is:
 - Minimum—0
 - Maximum—4294967295
10. **deny-period**—Set the length of time an entry is posted on the deny list. The host is deleted from the deny list after this time period. The default value is 30. A value of 0 disables the parameter. The valid range is:
 - Minimum—0
 - Maximum—4294967295
11. **nat-trust-threshold**—Enter maximum acceptable number of untrusted endpoints allowed before the Net-Net SBC demotes the entire NAT device to untrusted (dynamic demotion of NAT devices). The default is 0, meaning dynamic demotion of NAT devices is disabled.

The following example shows a host access policing configuration.

```
real m-config
  identifier          private
```

addr-prefix	192.168.200.0/24
network-interfaces	
private	0
dmz	disabled
internet	enabled
msm-reuse	disabled
qos-enabled	disabled
max-bandwidth	0
ext-policy-svr	
max-latency	0
max-jitter	0
max-packet-loss	0
observe-window-size	0
parent-real	
dns-real	
media-policy	
in-translational	
out-translational	
class-profile	
average-rate-limit	8000
access-control-trust-level	medium
invalid-signaling-threshold	200
maximum-signaling-threshold	0
untrusted-signaling-threshold	500
deny-period	30
symmetric-latching	disabled
pair-strip	disabled
trunk-context	

Configuring Overload Protection for Session Agents

The Net-Net SBC offers two methods to control SIP registrations to smooth the registration flow.

You can limit the:

- number of new register requests sent to a session agent (using the **max-register-sustain-rate** parameter)
- burstiness which can be associated with SIP registrations

The first method guards against the Net-Net SBC's becoming overwhelmed with register requests, while the second method guards against a transient registration that can require more than available registration resources.

SIP registration burst rate control allows you to configure two new parameters per SIP session agent—one that controls the registration burst rate to limit the number of new registration requests, and a second to set the time window for that burst rate. When the registration rate exceeds the burst rate you set, the 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.

To configure overload protection for session agents:

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

ACMEPACKET# **configure terminal**

2. Type **session-router** and press <Enter> to access the system-level configuration elements.

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

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

ACMEPACKET(session-router)# **session-agent**

ACMEPACKET(session-agent)#

4. **constraints**—Enable this parameter to set the sustained rate window constraint you configure in the next step. The default value is **disabled**. The valid values are:

- enabled | disabled

5. **sustain-rate-window**—Enter a number to set the sustained window period (in milliseconds) that is used to measure the sustained rate. (Refer to the max-sustain-rate information). The default value is zero (0). The valid range is:

- Minimum—0
- Maximum— $2^{32}-1$

6. **max-register-sustain-rate**—Enter a number to set the maximum number of registrations per second you want sent to the session agent. The default value is zero (0), disabling the parameter. The valid range is:

- Minimum—0
- Maximum—4294967295

7. **register-burst-window**—Define the window size in seconds for the maximum number of allowable SIP registrations. 0 is the minimum and default value for this parameter; the maximum value is 999999999.

8. **max-register-burst-rate**—Enter the maximum number of new registrations you want this session agent to accept within the registration burst rate window. If this threshold is exceeded, the Net-Net SBC will respond to new registration requests with 503 Service Unavailable messages. 0 is the minimum and default value for this parameter; the maximum value is 999999999.

9. Save and activate your configuration.

Media Policing

Media policing controls the throughput of individual media flows in the Net-Net SBC, which in turn provides security and bandwidth management functionality. The media policing feature works for SIP, H.323, SIP-H.323, and MGCP/NCS protocols. The media policing feature also lets you police static flows and RTCP flows.

The term media policing refers to flows that go through the Net-Net SBC. Flows that are directed to the host application are not affected by media policing.

You can use media policing to protect against two potential security threats that can be directed against your Net-Net SBC:

- Media DoS—Once media flows are established through the Net-Net SBC, 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 SBC 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 SBC watches for the codec specified in an SDP or H.245 message. When a match is made between the codec listed in an incoming session request and a configured **media-profile** configuration element, the Net-Net SBC 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 SBC will use the **media-profile** with the most permissive media policing constraints for all of the flows associated with the session. If a codec in the H.245/SDP message is not found in any configured **media-profile**, the Net-Net SBC 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 SBC on a media interface. The redirection is based on realm, source, destination, and protocol. When a flow matches the configured static flow criteria, besides being redirected toward a specified destination, its rate can also be controlled based on a static flow policing parameter found in the **static-flow** element. Static flow policing operates oblivious to the data contained within the flow.

Configuration Notes

Review the following information before configuring your Net-Net SBC 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 SBC unrestricted.

- RTP Policing
 - Set in the **media-profile** configuration element's **average-rate-limit** parameter to police RTP traffic with the CBR policing method.
 - **average-rate-limit**—Establishes the maximum speed for a flow in bytes per second.
- RTCP Policing
 - Set in the **media-manager-config** configuration element's **rtcp-rate-limit** parameter to police RTCP traffic with the CBR policing method.
 - **rtcp-rate-limit**—Establishes the maximum speed for an RTCP flow in bytes per second.

Static Flow Policing

Static flow policing is configured with one parameter found in the **static-flow** configuration element. To configure CBR, you have to set the **average-rate-limit**

parameter to a non-zero value. Setting the parameter listed below to 0 disables static flow policing, effectively letting the flow pass through the Net-Net SBC unrestricted.

In a CBR configuration, the **average-rate-limit** parameter determines the maximum bandwidth available to the flow.

- **average-rate-limit**—Establishes the maximum speed for a static flow in bytes per second.

Note: Static flow policing is not necessarily tied to any type of media traffic, it can affect flows of any traffic type.

Configuring Media Policing for RTP Flows

ACLI Instructions and Examples

You can configure media policing in the **media-profile** configuration element using the ACLI.

In the following example, you will configure media policing for the G723 media profile.

To configure media policing for RTP flows:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the **session-router** path.
ACMEPACKET(configure)# **session-router**
3. Type **media-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# **media-profile**
4. Select an existing media profile to which you will add policing constraints.
ACMEPACKET(media-profile)# **select**
<name>:
1: audio 4=G723 RTP/AVP 16 0 0 0

```
selection: 1
ACMEPACKET(media-profile)#

```

From this point, you can configure media policing parameters. To view all **media-profile** parameters, enter a ? at the system prompt

5. **average-rate-limit**—Enter the maximum rate in bytes per second for any flows that this **media-profile** polices. The default value is zero (0), disabling media policing. The valid range is:
 - Minimum—0
 - Maximum—125000000

Average rate limit values for common codecs:

- PCMU—80000 Bps
- G729—26000 Bps

The following example shows a **media-profile** configuration element configured for media policing.

```
media-profile
  name          G723

```

media-type	audio
payload-type	4
transport	RTP/AVP
req-bandwidth	16
frames-per-packet	0
parameters	
average-rate-limit	15000

Configuring 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. **rtp-rate-limit**—Enter the RTCP policing constraint in bytes per second. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—125000000

Configuring 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-real m; UDP

selection: 1

From this point, you can configure media policing parameters for static flows. To view all **static-flow** parameters, enter a ? at the system prompt

5. **average-rate-limit**—Enter the maximum rate in bytes per second for any flows that this **static-flow** polices. The default value is zero (0). The valid range is:

- Minimum—0
- Maximum—125000000

The following example shows a **static-flow** configuration element configured for media policing.

```
static-flow
  in-real-m-id          static-flow-in-real-m
  in-source              192.168.2.1/24
  in-destination        0.0.0.0
  out-real-m-id         static-flow-out-real-m
  out-source             192.168.128.1/24
  out-destination       0.0.0.0
  protocol               UDP
  average-rate-limit     15000
```

RTP Payload Type Mapping

The Net-Net SBC maintains a default list of RTP payload types mapped to textual encoding names as defined in RFC 3551.

The following table defines the preconfigured payload type for standard encodings.

Payload Type	Encoding Name	Audio (A) / Video (V)	Clock Rate
0	PCMU	A	8000
4	G723	A	8000
8	PCMA	A	8000
9	G722	A	8000
15	G728	A	8000
18	G729	A	8000

If you configure any payload type to encoding name mappings, the default mappings will be ignored. You must then manually enter all payload type mappings you use in the **media-profile** configuration element.

ITU-T to IANA Codec Mapping

The Net-Net SBC maintains a list of ITU-T (H.245) codecs that map to IANA RTP codecs. An ITU codec is directly mapped to an IANA Encoding Name for media profile lookups. All codecs are normalized to IANA codec names before any matches are made. New ITU-T codecs can not be added to the media profiles list.

The following table defines the ITU-T to IANA codec mappings.

ITU-T	IANA
g711Ulaw64k	PCMU
g711Alaw64k	PCMA

ITU-T	IANA
g726	G726
G7231	G723
g728	G728
g729wAnnexB	G729
g729	G729 fntp:18 annexb=no
H261VideoCapability	H261
H263VideoCapability	H263
t38Fax	T38

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**. The valid values are:
 - enabled | disabled
5. Save and activate your configuration.

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 and H.323 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.

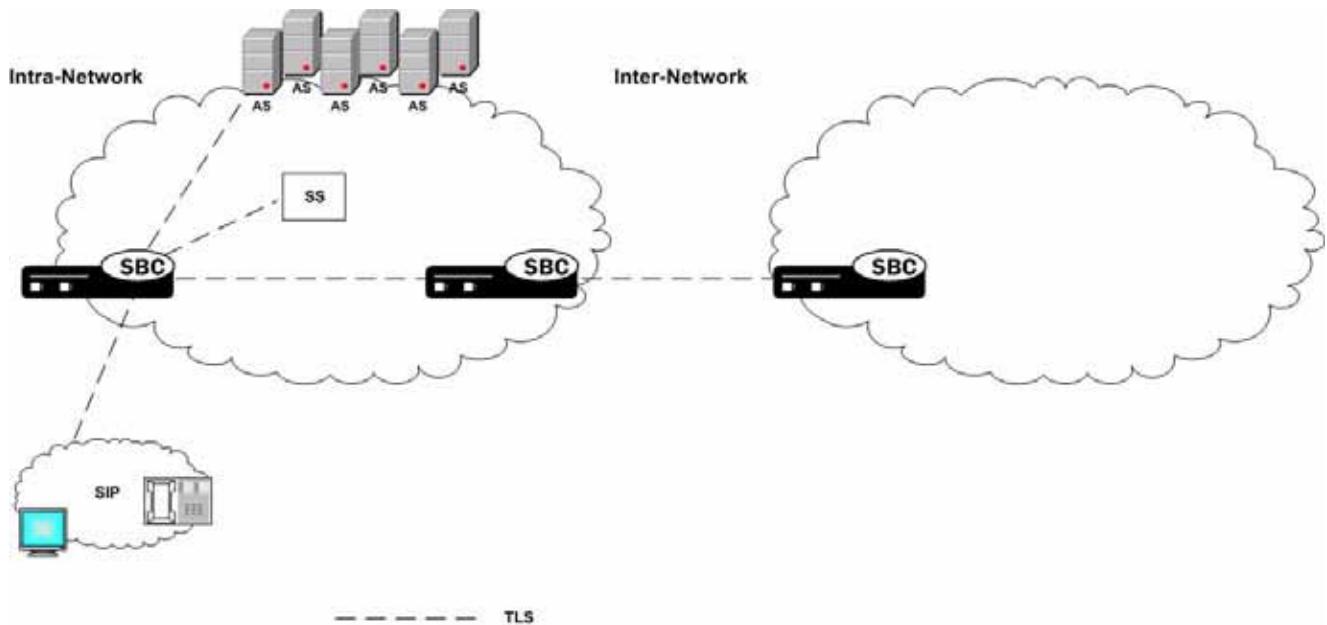
Configuring TCP SYN Attack Prevention

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 SBC chassis given the installation of the necessary bootloader and minimum hardware revision levels.

With the requisite hardware revision levels, the plug-on unit can be added to your Net-Net SBC 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 visible if the system's chassis cover is opened. New Net-Net SBCs outfitted with the SSM card will have the code labels already affixed in all required locations.

TLS support will not behave in the manner described here if you do not have the SSM component installed on your Net-Net SBC, because it is the presence of this hardware that enables the TLS software support.

The accelerator card performs:

- RSA

- Diffie-Hellman
- DES
- 3DES
- 40/128 bit ARCFOUR
- AES256
- Random number generation

TLS Features

The Net-Net SBC supports the following TLS features:

- TLSv1/SSLv3
- RFC 3261 specific SIPS and TLS support in SIP
- Importing X509v3 certificates in PKCS-7/X509v3 PEM/Base64 format
- Generating a private key and a certificate request in PKCS-10 PEM/Base64 format
- Displaying imported certificates in text format
- Configuration verification, including verification that all dependencies are resolved
- Connection reuse draft (draft-ietf-sip-connect-reuse-03.txt)
- HA for TLS—When the active system in an HA node fails, the standby has the same TLS-related configuration, which is accomplished through configuration checkpointing as described in the *HA Nodes* chapter.
 - Existing active calls are not affected by a failover—Enduser experiences no interruption or disturbance in service. SIP signaling messages sent over the connection following failover do not impact the active call.
 - New calls, new TLS connections are made

The 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, DES and ARC4 (40 and 128 bit) algorithms. It also supports:

- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA

- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_RSA_WITH_DES_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- ALL [default]
- NONE

Mapping SSL3 to TLSv1 Ciphers

The following table shows the mapping of SSL3 ciphers to TLSv1 ciphers:

SSL3	TLSv1
SSL_RSA_WITH_NULL_MD5	TLS_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA	TLS_RSA_WITH_NULL_SHA
SSL_RSA_WITH_RC4_128_MD5	TLS_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA	TLS_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA	TLS_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	TLS_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

Note: The 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 as because a request for TLS is controlled by the endpoint (TLS UA).

DoS Protection

The Net-Net SBC provides the following forms of 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 connections being requested by limiting the maximum number of connections for a SIP interface.
In other words, the maximum simultaneous TCP/TLS connections a SIP interface will allow in aggregate from all IP addresses served by that signaling interface.
 - Endpoints establishing TCP/TLS connections that never send any messages (application layer messages; once the TLS handshake completes).
This protection is triggered by inactivity, measured by lack of any message from a peer. The value specified for this timer is in seconds.
 - Endpoints requesting an initial registration that never send messages thereafter.
- Note:** It is expected that whenever an endpoint establishes a TCP/TLS connection, it will keep the connection active by sending additional messages or by using the NAT interval configuration. Whenever a connection is torn down because of inactivity, a log at the level "ERROR" is generated.
- Malformed packets by counting and limiting the maximum number of malformed packets.
Whenever the 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 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. 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 with the subject public key is used to verify digital signatures that provide a non-repudiation service protecting against the signing entity falsely denying some action, excluding certificate or CRL signing.
keyEncipherment (default with digitalSignature)	Used with the subject public key is used for key transport. (For example, when an RSA key is to be used for key management.)
dataEncipherment	Used with the subject public key is used for enciphering user data other than cryptographic keys.
keyAgreement	Used with the subject public key is used key agreement. (For example, when a Diffie-Hellman key is to be used for a management key.)
encipherOnly	The keyAgreement type must also be set.
decipherOnly	Used with the subject public key is used only for enciphering data while performing key agreement.
	The keyAgreement type must also be set.
	Used with the subject public key is used only for deciphering data while performing key agreement.

Extended Key Usage List

This section defines the values you can use in the **extended-key-usage-list** parameter.

Value	Description
serverAuth (default)	Used while the certificate is used for TLS server authentication. In 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 while 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 **generate-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 **import-certificate** 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
```

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 value is **US**.
6. **state**—Enter the name of the state of for the country. The default value is **MA**.
7. **locality**—Enter the name of the locality for the state. The default value is **Burlington**.
8. **organization**—Enter the name of the organization holding the certificate. The default value 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. The default value is **1024**. The valid range is:
 - 512 | 1024 | 2048
12. **alternate-name**—Enter the alternate name of the certificate holder.
13. **trusted**—Leave this parameters set to **enabled** to make the certificate trusted. Enter **disabled** to make this certificate untrusted. The default value is **enabled**. The valid values are:
 - enabled | disabled
14. **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](#) section.
15. **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](#) section.

Generating a Certificate Request

Using the ACLI **generate-certificate-request** command allows you to generate a private key and a certificate request in PKCS10 PEM format. You take this step once you have configured a certificate record.

The 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 **import-certificate** 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 access the internet through a browser such as Internet Explorer if it is available, or you can save the certificate request to a disk and then submit it to the CA.

To run the applicable command, you must use the value you entered in the name parameter of the certificate record configuration. You run the command from main Superuser mode command line:

```
ACMEPACKET# generate-certificate-request acmepacket
Generating Certificate Signing Request. This can take several
minutes...

-----BEGIN CERTIFICATE REQUEST-----
MIIDHzCCAoi gAwIBAgI AhMCUACEAHEwDQYJKoZI hvcNAQEFBQAwcDELMAkGA1UE
BhMCVVMxEzARBgNVBAgTCkNhGl mb3JuaWEExETAPBgNVBAcTCFNhbI BKb3NI MQ4W
DAYDVQQKEwVzaXBpdDEpMCCGA1UECxMgU2I waXQgVGVzdCBDZXJ0aWZpY2FOZSB
dXRob3JpdHkwHhcNMDUwNDEzMj EzNzQzWhcNMDgwNDEyMj EzNzQzWj BUMQswCQYD
VQGGEwJVUzELMAkGA1UECBMCTUExExARBgNVBAcTCkJ1cmxpbd0b24xFDASBgn
BAoTCOVuZ2I uZwVyaW5nMQ0wCwYDVQDwRhY21I MI GfMAOGCSqGSI b3DQEBAQUA
A4GNADCBi QKBgQCXj I e0yFKAUB3rKkKK/+59LT+rI GuW7Lgc1V6+hFTSr0co+ZsQ
bHFUWA15qXUUBTLJG13QN5VFG96f7gGAbWayf0S9Uymol d3JPCUDoGgb2E7m8i u
vtq7gwj SeKNXAw/y7yWy/c04FmUD2U0pZXOCNI R3Mns50AxQmq0bNYDhawI DAQAB
o4HdMI HaMBEGA1UdEQ0KMAi CBnBrdW1hcj AJBgvNVRMEAj AAMB0GA1UdDgQWBBTG
tpoda6Kmmn04L3Kg62t8BZJHTCmgYDVROj BI GSMI GPgBRrRhcU6pR2JYBUbhNU
2qHj VBShtqF0pHI wcDELMAkGA1UEBhMCVVMxExARBgNVBAgTCkNhGl mb3JuaWE
ETAPBgNVBAcTCFNhbI BKb3NI MQ4wDAYDVQQKEwVzaXBpdDEpMCCGA1UECxMgU2I w
aXQgVGVzdCBDZXJ0aWZpY2FOZSBdXRob3JpdHmCAQAwDQYJKoZI hvcNAQEFBQAD
gYEAbEs8nUCi +cA2hC/I M49Si tvh8QmpL81KONApsoC4Em24L+DZwz3uI noWj bj J
QhefcUfteNYkbuMH7LAK0hnDPvW+St4rQGVK6LjhZj 7/yeLXmYWI PUY3Ux40GVrd
2UgV/B2S0qh9NF+FQ+mNZ0I L7EuF4I xSz9/69LuYI XqKsG4=
-----END CERTIFICATE REQUEST-----;
```

WARNING: Configuration changed, run "save-config" command.

```
ACMEPACKET# save-config
Save-config received, processing.
waiting 1200 for request to finish
Request to 'SAVE-CONFIG' has finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot-activate'
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting 12000 for request to finish
Add LI flows
Li SysCl i entMgr:: handleNotifyReq
H323 Active Stack Cnt: 0
Request to 'ACTIVATE-CONFIG' has finished
Activate Complete
ACMEPACKET#
```

Importing a Certificate Using the ACLI

For an end-entity certificate, once a certificate is generated using the ACLI generate-certificate-request command, that request should be submitted to a CA for generation of a certificate in PKCS7 or X509v3 format. When the certificate has been generated, it can be imported into the Net-Net SBC using the **import-certificate** command.

The syntax is:

```
ACMEPACKET # import-certificate [try-all |pkcs7|x509] [certificate-
record file-name]
```

To import a certificate:

- When you use the **import-certificate** command, you can specify whether you want to use PKCS7 or X509v3 format, or try all. In the command line, you enter the command, the format specification, and the name of the certificate record.

```
ACMEPACKET# import-certificate try-all acme
```

The following will appear:

```
Please enter the certificate in the PEM format.  
Terminate the certificate with ";" to exit.....
```

```
-----BEGIN CERTIFICATE-----
```

```
MI IDHzCCAoi gAwIBAgI AhMCUACEAHEwDQYJKoZI hvcNAQEFBQAwcDELMAKGA1UE  
BhMCVVMxEzARBgNVBAgTCkNhGImb3JuaWEExETAPBgNVBAcTCFNhbIBKb3NI MQ4W  
DAYDVQQKEwVzaXBpdDEpMCCGA1UECxMgU2I waXQgVGVzdCBDZXJ0aWZpY2F0ZSB  
dXRob3JpdHkwHhcNMDUwNDEzMj EzNzQzWhcNMdgwNDEyMj EzNzQzWj BUMQswCQYD  
VQQGEwJVUzELMAKGA1UECBMCTUExExARBgNVBAcTCkJ1cmxpbd0b24xFDASBgNV  
BAoTCOVuZ2I uZWVyaW5nMQ0wCwYDVQQDEwRhY21I MI GfMA0GCSqGSI b3DQEBAQUA  
A4GNADCBi QKBgQCXj I e0yFKAUB3rKkKK/+59LT+rI GuW7Lgc1V6+hFTSr0co+ZsQ  
bHFUAA15qXUUBTLJG13QN5VFG96f7gGAbWayfOS9Uymol d3JPCUDoGgbE7m8i u  
vtq7gwj SeKNXAw/y7yWy/c04FmUD2U0pZXOCNI R3Mns50AxQmqObNYDhawI DAQAB  
o4HdMI HaMBEGA1UdEQQKMAi CBnBrdW1hcj AJBgvNHRMEAj AAMBOGA1UdDgQWBTTG  
tpodxa6Kmmn04L3Kg62t8BZJHTCBmgYDVROj BI GSMI GPgBRrRhcU6pR2JYBUbhNU  
2qHj VBShtqFOpHI wcDELMAKGA1UEBhMCVVMxEzARBgNVBAgTCkNhGImb3JuaWE  
ETAPBgNVBAcTCFNhbIBKb3NI MQ4wDAYDVQQKEwVzaXBpdDEpMCCGA1UECxMgU2I w  
aXQgVGVzdCBDZXJ0aWZpY2F0ZSBdXRob3JpdHmCAQAwDQYJKoZI hvcNAQEFBQAD  
gYEAbEs8nUCi +cA2hC/I M49Si tvh8QmpL81KONApsoC4Em24L+DZwz3uI noWj bj J  
QhefcUfteNYkbuMH7LAK0hnDPvW+St4rQGVK6LJhZj 7/yeLXmYWI PUY3Ux40GVrd  
2UgV/B2S0qH9NF+FQ+mNZOI L7EuF4I xSz9/69LuYI XqKsG4=
```

```
-----END CERTIFICATE-----;
```

Certificate imported successfully....

WARNING: Configuration changed, run "save-config" command.

- Save your configuration.

```
ACMEPACKET# save-config
```

```
Save-Config received, processing.  
waiting 1200 for request to finish  
Request to 'SAVE-CONFIG' has finished,  
Save complete  
Currently active and saved configurations do not match!  
To sync & activate, run 'activate-config' or 'reboot activate'.
```

- Synchronize and activate your configurations.

```
ACMEPACKET# activate-config
```

```
Activate-Config received, processing.  
waiting 120000 for request to finish  
Add LI Flows  
Li SysCl ientMgr:: handleNotifyReq  
H323 Active Stack Cnt: 0  
Request to 'ACTIVATE-CONFIG' has finished,  
Activate Complete  
ACMEPACKET#
```

Importing a Certificate Using FTP

You can also put the certificate file in the directory /ramdrv and then executing the **import-certificate** command or by pasting the certificate in the PEM/Base64 format into the ACLI. If you paste the certificate, you might have to copy and paste it a portion at a time rather than pasting in the whole thing at once.

To import the certificate using FTP:

1. FTP the certificate file on to the Net-Net SBC (directory /ramdrv), let us say the name of the certificate file is `cert.pem`.
2. Once the certificate is successfully transferred to the Net-Net SBC, run the **import-certificate** command.

The syntax is:

```
ACMEPACKET# import-certificate [try-all |pkcs7|x509] [certificate-record file-name]
```

Using the command will look like this when you have used FTP.

```
ACMEPACKET# import-certificate try-all acme cert.pem
Certificate imported successfully...
WARNING: Configuration changed, run "save-config" command.
```

4. Save your configuration.

```
ACMEPACKET# save-config
Save-Config received, processing.
waiting 1200 for request to finish
Request to 'SAVE-CONFIG' has finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.
```

5. Synchronize and activate your configurations.

```
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting 120000 for request to finish
Add LI Flows
Li SysCl i entMgr::handl eNoti fyReq
H323 Active Stack Cnt: 0
Request to 'ACTIVATE-CONFIG' has finished,
Activate Complete
ACMEPACKET#
```

Configuring a TLS Profile

The TLS profile configuration has been added to the security section of the ACLI's configure terminal menu. This configuration holds the information required to run SIP over TLS.

In the ACLI menu for this configuration, the parameters appear as follows:

<code>tls-profile</code>	
<code>name</code>	tls profile name
<code>end-entity-certificate</code>	end entity certificate for the TLS connection
<code>trusted-ca-certificates</code>	list of trusted certificate records
<code>cipher-list</code>	list of ciphers
<code>verify-depth</code>	maximum length of the certificate chain
<code>mutual-authenticate</code>	mutually authenticate

To configure a TLS profile:

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# config 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](#) section of this chapter for more details.
8. **verify-depth**—Specify the maximum depth of the certificate chain that will be verified. The default value is **10**. The valid range is:
 - Minimum—0
 - Maximum—10
9. **mutual-authenticate**—Define whether or not you want the Net-Net SBC to mutually authenticate the client. The default value is **disabled**. The valid values are:
 - enabled | disabled
10. **tls-version**—Enter the TLS version you want to use with this TLS profile. Valid values are **TLSv1**, **SSLv3**, and **compatibility** (default).
11. Save your work.
12. Exit out to the configuration terminal menu to apply the TLS profile.
ACMEPACKET(tls-profile)# exit
ACMEPACKET(security)# exit
ACMEPACKET(configure)#

Applying a TLS Profile

To apply the TLS profile, you need to specify it for the SIP interface with which it will be used. You must take this step from within the SIP interface configuration.

1. Type **session-router** and press <Enter> to access the **session-router** path.
ACMEPACKET(configure)# session-router
2. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
3. Select the existing SIP interface to which you want to apply the TLS profile. If you do not know the 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. **transport-protocol**—Change the transport protocol to TLS.

```
ACMEPACKET(sip-interface)# transport-protocol tls
```

6. **tls-profile**—Enter the name of the TLS profile you want applied. This is the same value you enter for the name parameter in the TLS profile configuration. This profile will be applied when the transport protocol is TLS.

```
ACMEPACKET(sip-interface)# tls-profile acmepacket
```

7. Save your updated SIP interface configuration.

Reusing a TLS Connection

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

Brief Version

Obtaining the brief version uses this syntax, and will appear like the following example:

```
ACMEPACKET# show security certificates brief acmepacket
```

```
certificate-record: acmepacket
```

```
Certificate:
```

```
Data:
```

```
Version: 3 (0x2)
```

```

Serial Number:
02:13:02:50:00:84:00:71
Issuer:
C=US
ST=California
L=San Jose
O=sipit
OU=Sipit Test Certificate Authority
Subject:
C=US
ST=MA
L=Burlington
O=Engineering
CN=acme
ACMEPACKET#

```

Detailed Version

Obtaining the detailed version uses this syntax, and will appear like the following example:

```

ACMEPACKET# show security certificates detail acmepacket

certificate-record: acmepacket

Certificate:
Data:
Version: 3 (0x2)
Serial Number:
02:13:02:50:00:84:00:71
Signature Algorithm: sha1WithRSAEncryption
Issuer:
C=US
ST=California
L=San Jose
O=sipit
OU=Sipit Test Certificate Authority
Validity
Not Before: Apr 13 21:37:43 2005 GMT
Not After : Apr 12 21:37:43 2008 GMT
Subject:
C=US
ST=MA
L=Burlington
O=Engineering
CN=acme
X509v3 extensions:
X509v3 Subject Alternative Name:
DNS: pkumar
X509v3 Basic Constraints:
CA: FALSE

ACMEPACKET#

```

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 default value is zero (0). The default disables the parameter. The valid range is:
 - Minimum—0
 - Maximum—40000
5. **per-src-ip-max-incoming-conns**—Enter the maximum number of connections allowed from an end point. The default value is zero (0). The default disables the parameter. The valid range is:
 - Minimum—0
 - Maximum—40000
- Note:** To make this parameter effective, you need to set the realm's access-control-trust-level to low or medium.
6. **inactive-conn-timeout**—Enter the time in seconds you want a connection from an endpoint discontinued. This provides protection from end points doing an initial registration without sending any messages. The default value is zero (0). The default disables the parameter. The valid range is:
 - Minimum—0
 - Maximum—999999999
7. Save and activate your configuration.

Configuring the SIP Configuration

To configure the SIP configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.
ACMEPACKET(configure)# **session-router**
3. Type **sip-config** and press <Enter>. The system prompt changes.
ACMEPACKET(session-router)# **sip-config**
ACMEPACKET(sip-config)#

From this point, you can configure SIP configuration parameters. To view all sip-config parameters, enter a ? at the system prompt.
4. **inactive-dynamic-conn**—Enter the time in seconds after which 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 default value is 32. The valid range is:
 - Minimum—1
 - Maximum—999999999

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>.
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(real-m-config)#
From this point, you can configure realm parameters. To view all realm configuration parameters, enter a ? at the system prompt.
4. **deny-period**—Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is **30**. The valid range is:
 - Minimum—0
 - Maximum—4294967295
5. **invalid-signal-threshold**—Enter the maximum number of simultaneous TCP/TLS connections for this SIP interface. The default value is **0**. The valid range is:
 - Minimum—0
 - Maximum—4294967295

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 default value is **none**. The valid values are:
 - **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list.
 - **low**—Host can be promoted to the trusted list or demoted to the deny list.
 - **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list.
 - **high**—Host is always trusted.
7. Save and activate your configuration.

TLS Session Caching

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-global` and press <Enter>.

ACMEPACKET(security)# **tls-global**
ACMEPACKET(tls-global)#
4. **session-caching**—Set the state for TLS session caching to **enabled** if you want to turn this feature on. The default value is **disabled**. The valid values are:
 - `enabled` | `disabled`
5. **session-cache-timeout**—Enter the time in hours that you want the Net-Net SBC to cache unique session identifiers so that previously authenticated clients can reconnect. The default value is **12**. A value of **0** disables this parameter. The valid range is:
 - Minimum—**0**
 - Maximum—**24**

If you set this parameter to 0, then cache entries will never age (and not be deleted from the cache unless you use the `clear-cache tls` command to delete all entries from the TLS cache). RFC 2246, “The TLS Protocol Version 1.0,” recommends that you set this parameter at the maximum, 24.

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# config terminal
2. Type **session-router** and press <Enter> to access the signaling-level configuration elements.
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
3. Type **sip-interface** and press <Enter>.
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
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 value is **0**, which will not start the timer. The valid range is:
 - Minimum—0
 - Maximum—999999999
5. Save and activate your configuration.

Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) is defined in RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. The protocol enables users to determine the revocation state of a specific certificate, and may provide a more efficient source of revocation information than is possible with Certificate Revocation Lists (CRL).

The protocol specifies the data exchanged between an OCSP client (for example, the Net-Net SBC) and an OCSP responder, the Certification Authority (CA), or its delegate, that issued the target certificate. An OCSP client issues a request to an OCSP responder and suspends acceptance of the certificate in question until the responder replies with a certificate status.

Certificate status is reported as

- good
- revoked
- unknown

good indicates a positive response to the status inquiry. At a minimum, this positive response indicates that the certificate is not revoked, but does not necessarily mean that the certificate was ever issued or that the time at which the response was produced is within the certificate's validity interval.

revoked indicates that the certificate has been revoked, either permanently or temporarily.

unknown indicates that the responder cannot identify the certificate.

Caveats

OCSP is currently supported only on TLS interfaces; it is not currently supported for use with IKEv1 and IKEv2.

ACLI Instructions and Examples

OCSP configuration consists of

1. Configuring one or more certificate status profiles; each profile contains information needed to contact a specific OCSP responder.
2. Enabling certificate revocation checking by assigning a certificate status profile to a previously configured TLS profile.

To create a certificate status profile:

1. From superuser mode, use the following command sequence to access *cert-status-profile* configuration mode. While in this mode, you provide the information required to access one or more OCSP responders.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# cert-status-profile
ACMEPACKET(cert-status-profile)#

```

2. Use the required **name** parameter to identify this *cert-status-profile* instance — each profile instance provides configuration data for a specific OCSP responder. **name** is used to distinguish between multiple profile instances.

3. Use the required **ip-address** parameter to specify the IPv4 address of the OCSP responder.
4. Use the optional **port** parameter to specify the destination port.
In the absence of an explicitly configured value, the default port number of 80 is used.
5. Use the optional **realm-id** parameter to specify the realm used to transmit OCSP requests.
In the absence of an explicitly configured value, the default specifies service across the *wancom0* interface.
6. Use the optional **requester-cert** parameter only if OCSP requests are signed; ignore this parameter if requests are not signed.
RFC 2560 does not require signed requests; however, local or CA policies can mandate digital signature..
7. Use the required **responder-cert** parameter to identify the certificate used to validate OCSP responses — a public key of the OCSP responder.
RFC 2560 requires that all OCSP responders digitally sign OCSP responses, and that OCSP clients validate incoming signatures.
Provide the name of the certificate configuration element that contains the certificate used to validate the signed OCSP response.
8. Use the optional **retry-count** parameter to specify the maximum number of times to retry an OCSP responder in the event of connection failure.
If the retry counter specified by this parameter is exceeded, the OCSP requester either contacts another responder (if multiple responders have been configured within this *cert-status-profile*) and quarantine the unavailable responder for a period defined the **dead-time** parameter.
In the absence of an explicitly configured value (an integer within the range 0 through 10), the default of 1 is used.

```
ACMEPACKET(cert-status-profile)# retry-count 2
ACMEPACKET(cert-status-profile)#

```
9. Use the optional **dead-time** parameter to specify the quarantine period imposed on an unavailable OCSP responder.
In the absence of an explicitly configured value (an integer within the range 0 through 3600 seconds), the default value (0) is used.
Customer implementations utilizing a single OCSP responder are encouraged to retain the default value, or to specify a brief quarantine period to prevent lengthy service outages.
10. Retain default values for the **type** and **trans-protocol** parameter to specify OCSP over an HTTP transport protocol.
11. Use **done**, **exit**, and **verify-config** to complete configuration of this *cert-status-profile* instance.
12. Repeat Steps 1 through 11 to configure additional certificate status profiles.

To enable certificate status checking:

1. Move to *tls-profile* configuration mode.


```
ACMEPACKET# config terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)#

```
2. Use the required **cert-status-check** parameter to enable OCSP in conjunction with an existing TLS profile.
3. Use the required **cert-status-profile-list** parameter to assign one or more *cert-status-profiles* to the current TLS profile.

Each assigned *cert-status-profile* provides the information needed to access a single OCSP responder.

Use quotation marks to assign multiple OCSP responders. The following sequence assigns three *cert-status-profiles*, *VerisignClass3Designate*, *Verisign-1*, and *Thawte-1* to the TLS-1 profile.
4. Use **done**, **exit**, and **verify-config** to complete configuration.

Sample Configuration:

```
ACMEPACKET# config terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# cert-status-profile
ACMEPACKET(cert-status-profile)# name VerisignClass3Designate
ACMEPACKET(cert-status-profile)# ip-address 192.168.7.100
ACMEPACKET(cert-status-profile)# responder-cert
VerisignClass3ValidateOCSP
ACMEPACKET(cert-status-profile)# done
ACMEPACKET(cert-status-profile)# exit
...
...
ACMEPACKET# config terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tls-profile
ACMEPACKET(tls-profile)# select
<name>:
1. TLS-1
2. TLS-2
3. TLS-3

selection: 1
ACMEPACKET(tls-profile)# cert-status-check enabled
ACMEPACKET(cert-status-profile)# cert-status-profile-list
"VerisignClass3Designate Verisign-1 Thawte-1"
ACMEPACKET(cert-status-profile)# done
ACMEPACKET(cert-status-profile)# exit
```

Key Exchange Protocols

Key exchange protocols enable secure communications over an untrusted network by deriving and distribution shared keys between two or more parties. The Internet Key Exchange (IKEv1) Protocol, originally defined in RFC 2409, provides a method for creating keys used by IPsec tunnels. MIKEY defined in RFC 3880, and Session Description Protocol Security Descriptions for Media Streams (SDES), defined in RFC 4568, provide alternative methods for creating keys used to encrypt Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP) transactions.

Each of these protocols is described in the following sections.

IKEv1 Protocol

IKEv1 is specified by a series of RFCs, specifically RFCs 2401 through 2412. The most relevant are:

- RFC 2407, *The Internet IP Security Domain of Interpretation for ISAKMP*
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*
- RFC 2409, *The Internet Key Exchange (IKE)*
- RFC 2412, *Oakley Key Determination Protocol*

IKEv1 combines features of the Internet Security Association and Key Management Protocol (ISAKMP) and Oakley Key Determination Protocol in order to negotiate Security Associations (SA) for two communicating peers. IKEv1 also provides for key agreement using Diffie-Hellman.

IKEv1 uses two phases. Phase 1 is used to establish an ISAKMP Security Association for IKEv1 itself. Phase 1 negotiates the authentication method and symmetric encryption algorithm to be used. Phase 1 requires either six messages (main mode) or three messages (aggressive mode).

Phase 2 negotiates the SA for two IPsec peers and is accomplished with three messages.

The initial IKEv1 implementation supports RFC 2409, *Internet Key Exchange*, and RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*.

ACLI Instructions and Examples

IKEv1 configuration consists of five steps.

1. Configure IKEv1 global parameters.
2. Optionally, enable and configure Dead Peer Detection (DPD) Protocol.
3. Configure IKEv1 interfaces.
4. Configure IKEv1 Security Associations (SA).
5. Assign the IKEv1 SA to an IPsec Security Policy.

IKEv1 Global Configuration

To configure global IKEv1 parameters:

- From superuser mode, use the following command sequence to access *ike-config* configuration mode. While in this mode, you configure global IKEv1 configuration parameters.

```
ACMEPACKET# config urre termi nal  
ACMEPACKET(configure)# securi ty  
ACMEPACKET(securi ty)# i ke  
ACMEPACKET(i ke)# I ke-confi g  
ACMEPACKET(i ke-confi g)#
```

2. Use the **ike-version** parameter to specify IKEv1.
Use 1 to specify IKEv1 operations.
 3. Use the **log-level** parameter to specify the contents of the IKEv1 log.
Events are listed below in descending order of criticality.

- emergency (most critical)
critical
major
minor
warning
notice
info (least critical –
trace (test/debug, n
debug (test/debug, n
detail (test/debug, n

In the absence of an explicitly configured value, the default value of *info* is used.

4. Use the optional **udp-port** parameter to specify the port monitored for IKEv1 protocol traffic.

In the absence of an explicitly configured value, the default port number of 500 is used.

5. Use the optional **negotiation-timeout** parameter to specify the maximum interval (in seconds) between Diffie-Hellman message exchanges.

In the absence of an explicitly configured value, the default specifies a 15 second timeout value.

6. Use the optional **event-timeout** parameter to specify the maximum time (in seconds) allowed for the duration of an IKEv1 event, defined as the successful establishment of an IKE or IPsec Security Association (SA).

In the absence of an explicitly configured value, the default specifies a 60 second time span.

7. Use the optional **phase1-mode** parameter to specify the IKE Phase 1 exchange mode.

During Phase 1 the IKE initiator and responder establish the IKE SA, using one of two available methods.

main mode — (the default) is more verbose, but provides greater security in that it does not reveal the identity of the IKE peers. Main mode requires six messages (3 requests and corresponding responses) to (1) negotiate the IKE SA, (2) perform a Diffie-Hellman exchange of cryptographic material, and (3) authenticate the remote peer.

aggressive mode — is less verbose (requiring only three messages), but less secure in providing no identity protection, and less flexible in IKE SA negotiation.

In the absence of an explicitly configured value, the default (*main mode*) is used.

8. Use the optional **phase1-dh-mode** parameter to specify the Diffie-Hellman Group used during IKE Phase 1 negotiation.
 - dh-group1* — as initiator, propose Diffie-Hellman group 1 (768-bit primes, less secure)
 - dh-group2* — as initiator, propose Diffie-Hellman group 2 (1024-bit primes, more secure)
 - first-supported* — (the default) as responder, use the first supported Diffie-Hellman group proposed by initiator
9. If functioning as the IKE initiator, use the optional **phase1-life-seconds** parameter to specify the proposed lifetime (in seconds) for the IKE SA established during IKE Phase 1 negotiations.

Allowable values are within the range 1 through 999999999 (seconds) with a default of 3600 (1 hour).

This parameter can safely be ignored if functioning as a IKE responder.
10. If functioning as the IKE responder, use the optional **phase1-life-seconds-max** parameter to specify the maximum time (in seconds) accepted for IKE SA lifetime during IKE Phase 1 negotiations.

Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (1 day).

This parameter can safely be ignored if functioning as a IKE initiator.
11. If functioning as the IKE initiator, use the optional **phase2-life-seconds** parameter to specify the proposed lifetime (in seconds) for an IPsec SA established during IKE Phase 2 negotiations.

Allowable values are within the range 1 through 999999999 (seconds) with a default of 28800 (8 hours).

This parameter can safely be ignored if functioning as a IKE responder.
12. If functioning as the IKE responder, use the optional **phase2-life-seconds-max** parameter to specify the maximum time (in seconds) accepted for IPsec SA lifetime during IKE Phase 2 negotiations.

Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (1 day).

This parameter can safely be ignored if functioning as a IKE initiator.
13. Use the optional **phase2-exchange-mode** parameter to specify the Diffie-Hellman group used in Phase 2 negotiations.
 - dh-group1* — use Diffie-Hellman group 1 (768-bit primes, less secure)
 - dh-group2* — use Diffie-Hellman group 2 (1024-bit primes, more secure)
 - no-forward-secrecy* — use the same key as used during Phase 1 negotiation

Note: Forward security indicates that compromise of a single key permits access only to data encrypted with that specific key. Failure to generate a new key for IKE Phase 2 potentially compromises additional data.

phase1-group — (the default) use the same Diffie-Hellman group as used during Phase 1 negotiation

14. Use the **shared-password** parameter to specify the PSK (pre-shared key) used during authentication with the remote IKE peer.
The PSK is a string of ACSII printable characters no longer than 255 characters (not displayed by the ACLI).
This global PSK can be over-ridden by an interface-specific PSK.
15. Use the optional **dpd-time-interval** parameter to specify the maximum period of inactivity before the DPD protocol is initiated on a specific endpoint.
Allowable values are within the range 1 through 999999999 (seconds) with a default of 0.
The default value, 0, disables the DPD protocol; setting this parameter to a non-zero value globally enables the protocol and sets the inactivity timer.
16. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv1 global parameters instance.

DPD Protocol Configuration

If you enabled the DPD protocol with the **dpd-time-interval** parameter, use the following procedure to create a DPD template, an operational set of DPD parameters, that you subsequently assign to one or more IKEv1 interfaces.

To configure DPD parameters:

1. From superuser mode, use the following command sequence to access *dpd-params* configuration mode. While in this mode, you configure DPD templates.


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# ike
ACMEPACKET(ike)# dpd-params
ACMEPACKET(dpd-params)#

```
2. Use the required **name** parameter to provide a unique identifier for this *dpd-params* instance.
name enables the creation of multiple *dpd-params* instances.
3. Use the **max-loop** parameter to specify the maximum number DPD peers examined every dpd-interval, whose value is established during IKv1 global configuration.
If CPU workload surpasses the threshold set by **max-cpu-limit**, this value is over-ridden by **load-max-loop**.
Allowable values are within the range 1 through 999999999 (endpoints) with a default of 100.
4. Use the **max-endpoints** parameter to specify the maximum number of simultaneous DPD protocol negotiations supported when the CPU is not under load (as specified by the **max-cpu-limit** property).
If CPU workload surpasses the threshold set by **max-cpu-limit**, this value is over-ridden by **load-max-endpoints**.
Allowable values are within the range 1 through 999999999 (endpoints) with a default of 25.

5. Use the **max-cpu-limit** parameter to specify a threshold value (expressed as a percentage of CPU capacity) at which DPD protocol operations are minimized to conserve CPU resources.
Allowable values are within the range 0, which effectively disables DPD operations, through 100 (percent) with a default of 60.
6. Use the **load-max-loop** parameter to specify the maximum number of endpoints examined every **dpd-time-interval** when the CPU is under load, as specified by the **max-cpu-limit** parameter.
Allowable values are within the range 1 through 999999999 (endpoints) with a default of 40. Ensure that the configured value is less than the value assigned to **max-loop**.
7. Use the **load-max-endpoints** parameter to specify the maximum number of simultaneous DPD Protocol negotiations supported when the CPU is under load, as specified by the **max-cpu-limit** property.
Allowable values are within the range 1 through 999999999 (endpoints) with a default of 5. Ensure that the configured value is less than the value assigned to **max-endpoints**.
8. Use **done**, **exit**, and **verify-config** to complete configuration of the DPD template instance.
9. Repeat Steps 1 through 8 to configure additional DPD templates.

IKEv1 Interface Configuration

To configure IKEv1 interface parameters:

1. From superuser mode, use the following command sequence to access *ike-config* configuration mode. While in this mode, you configure IKEv1 interface parameters.


```
ACMEPACKET# config terminal
ACMEPACKET(config)# security
ACMEPACKET(security)# ike
ACMEPACKET(ike)# ike-interface
ACMEPACKET(ike-interface)#

```
2. Use the **address** parameter to specify the IPv4 address of the interface.
3. Use the **realm-id** parameter to specify the realm that contains the IP address assigned to this IKEv1 interface.
4. Use the **ike-mode** parameter to specify the operational mode, either *responder* (the default) or *initiator*.
5. If DPD has been enabled at the global level, use the **dpd-params-name** parameter to assign a DPD template, an operational set of DPD parameters, to the current IKEv1 interface.

If DPD has not been enabled, this parameter can be safely ignored.
6. Use the optional **shared-password** parameter to assign an interface PSK.
This IKEv1-interface-specific value over-rides the global default value set at the IKE configuration level.
7. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv1 interface.
8. Repeat Steps 1 through 7 to configure additional IKEv1 interfaces.

IKEv1 Security Association Configuration

An IKEv1 SA identifies cryptographic material available for IPsec tunnel establishment.

To configure IKEv1 SA parameters:

- From superuser mode, use the following command sequence to access *ike-sainfo* configuration mode. While in this mode, you configure global IKEv1 SAs.

```
ACMEPACKET# configuration terminal
ACMEPACKET(configuration)# security
ACMEPACKET(security)# ike
ACMEPACKET(ike)# ike-sainfo
ACMEPACKET(ike-sainfo)#

```

- Use the required **name** parameter to provide a unique identifier for this *ike-sainfo* instance.

name enables the creation of multiple *ike-sainfo* instances.

- Use the **security-protocol** parameter to specify the IPsec security (authentication and encryption) protocols supported by this SA.

The following security protocols are available.

Authentication Header (AH) — the default value — as defined by RFC 4302, *IP Authentication Header*, which provides authentication integrity to include the mutual identification of remote peers, non-repudiation of received traffic, detection of data that has been altered in transit, and detection of data that has been replayed, that is copied and then re-injected into the data stream at a later time. Authentication services utilize the authentication algorithm specified by the **auth-algo** property.

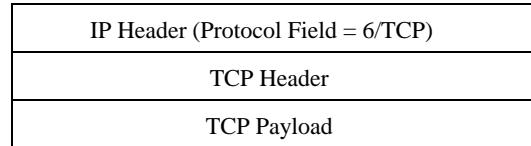
Encapsulating Security Payload (ESP) as defined by RFC 4303, *IP Encapsulating Security Payload*, which provides both authentication and privacy services. Privacy services utilize the encryption algorithm specified by the **encryption-algo** property.

ESP-AUTH (also RFC 4303-based), which supports ESP's optional authentication.

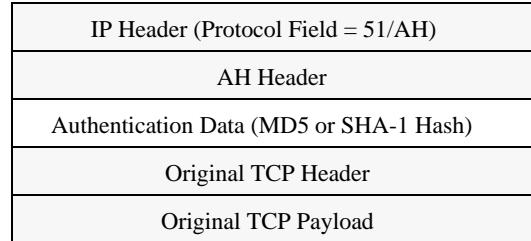
ESP-NULL (also RFC 4303-based) which proves NULL encryption as described in RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*. This option provides no privacy services, and is not recommended for production environments.

Refer to the following figures for additional details.

Original IP Datagram

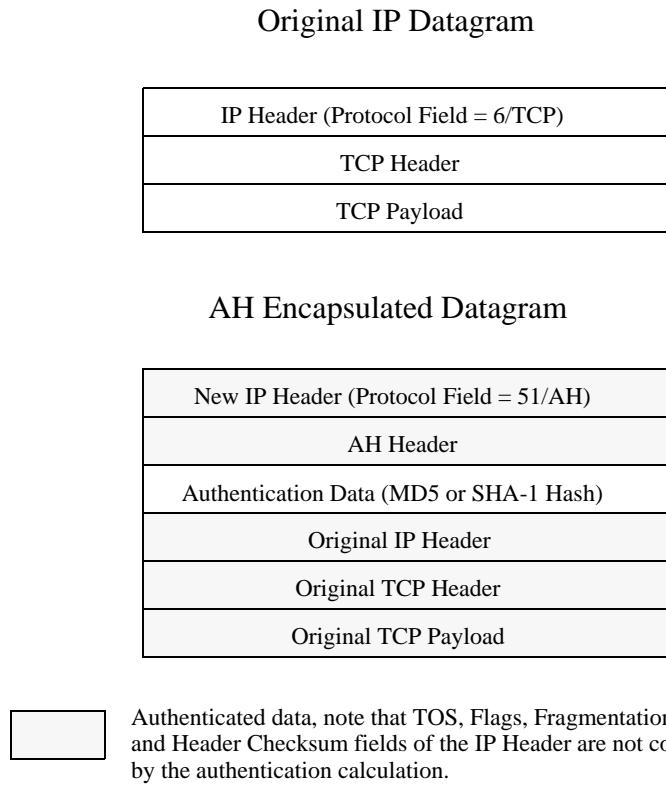
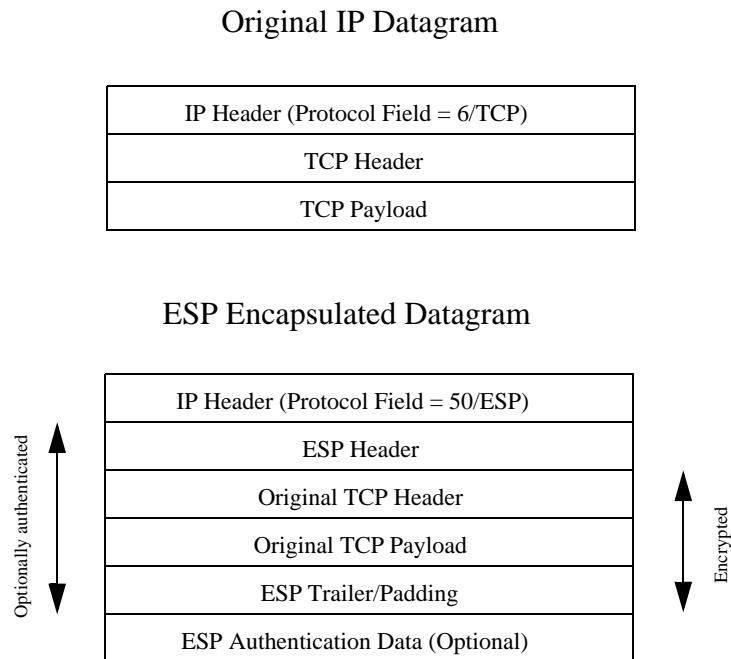


AH Encapsulated Datagram



Authenticated data, note that TOS, Flags, Fragmentation, TTL, and Header Checksum fields of the IP Header are not covered by the authentication calculation.

Figure 11: AH Transport Mode

**Figure 12: AH Tunnel Mode****Figure 13: ESP Transport Mode**

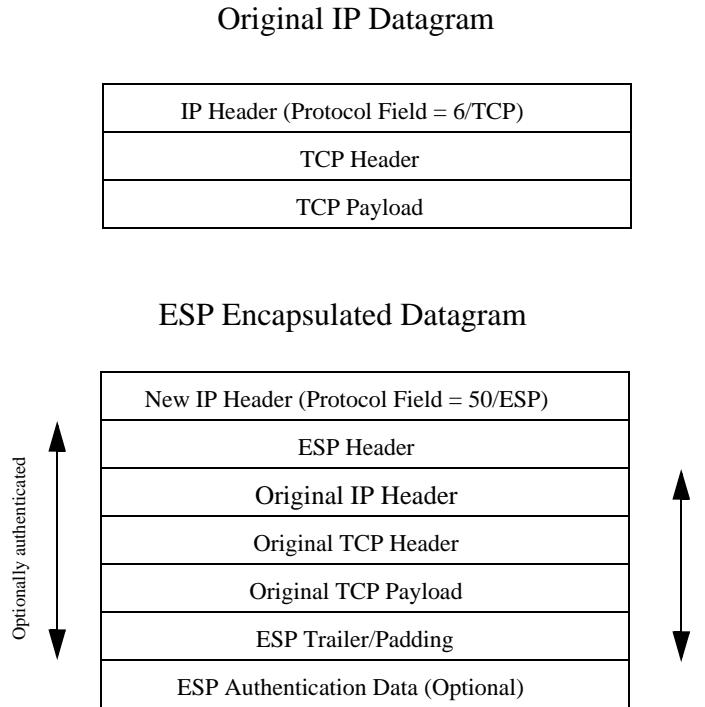


Figure 14: ESP Tunnel Mode

```
ACMEPACKET(i ke-sai nfo)# security-protocol esp
ACMEPACKET(i ke-sai nfo)#

```

4. Use the **auth-algo** parameter to specify the authentication algorithms supported by this SA.

The following authentication protocols are available

Message Digest Algorithm 5 (md5) — as defined by RFC 1321, *The MD5 Message-Digest Algorithm*.

Secure Hash Algorithm (sha) — as defined by FIPS PUB 180-1, *Secure Hash Standard*.

any (the default) — supports both MD5 and SHA-1.

```
ACMEPACKET(i ke-sai nfo)# auth-algo md5
ACMEPACKET(i ke-sai nfo)#

```

5. Use the **encryption-algo** parameter to specify the encryption algorithms supported by this SA.

The following encryption protocols are available

Triple DES (3des) — as defined by ANSI X.9.52 1998, *Triple Data Encryption Algorithm Modes of Operation*.

Advanced Encryption Standard (aes) — FIPS PUB 197, *Advanced Encryption Standard*.

NULL Encryption (null) — as described in RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*. This option provides no privacy services, and is not recommended for production environments.

any (the default) — supports all listed encryption protocols.

```
ACMEPACKET(i ke-sai nfo)# encrypti on-al go aes
ACMEPACKET(i ke-sai nfo)#

```

6. Use the **ipsec-mode** parameter to specify the IPSec operational mode.

Transport mode (the default) provides a secure end-to-end connection between two IP hosts. Transport mode encapsulates the IP payload.

Tunnel mode provides VPN service where entire IP packets are encapsulated within an outer IP envelope and delivered from source (an IP host) to destination (generally a secure gateway) across an untrusted internet.

Refer to the previous figures for encapsulation details.

```
ACMEPACKET(i ke-sai nfo)# ipsec-mode tunnel
ACMEPACKET(i ke-sai nfo)#

```

7. If **ipsec-mode** is *tunnel*, use the required **tunnel-local-addr** parameter to specify the IP address of the local IKEv1 interface that terminates the IPsec tunnel.

This parameter can safely be ignored if **ipsec-mode** is *transport*.

```
ACMEPACKET(i ke-sai nfo)# tunnel -l ocal -addr 192.169.204.14
ACMEPACKET(i ke-sai nfo)#

```

8. If **ipsec-mode** is *tunnel*, use the **tunnel-remote-addr** parameter to specify the IP address of the remote IKEv1 peer that terminates the IPsec tunnel.

Provide the remote IP address, or use the default wild-card value (*) to match all IP addresses.

This parameter can safely be ignored if **ipsec-mode** is *transport*.

```
ACMEPACKET(i ke-sai nfo)# tunnel -remote-addr *
ACMEPACKET(i ke-sai nfo)#

```

9. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv1 SA.

10. Repeat Steps 1 through 9 to configure additional IKEv1 SAs.

IPsec Security Policy Configuration

Use the following procedure to assign an IKEv1 SA to an existing IPsec Security Policy. Note that the network interface supported by the IPsec Security Policy must have been configured as an IKEv1 interface.

1. From superuser mode, use the following command sequence to access *ike-config* configuration mode. While in this mode, you configure global IKEv1 configuration parameters.

```
ACMEPACKET# config terminal
ACMEPACKET(config)# security
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)# security-policy#
ACMEPACKET(security-policy)#

```

2. Use the required **ike-sainfo-name** parameter to assign an IKV1 SA to this IPsec Security Policy.

3. Use **done**, **exit**, and **verify-config** to complete configuration of IPsec Security Policy.

SDP Session Description Protocol

The Secure Real-Time Transport Protocol, as described in RFC 3711, *The Secure Real-time Transport Protocol (SRTP)*, provides a framework for the encryption and authentication of Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) streams. Both RTP and RTCP are defined by RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*.

Encryption ensures that the call content and associated signalling remains private during transmission. Authentication ensures that (1) received packets are from the purported source, (2) packets are not been tampered with during transmission, and (3) a packet has not been replayed by a malicious server.

Protocol Overview

While the RFC 3711 framework provides encryption and authentication procedures and defines a set of default cryptographic transforms required for RFC compliance, it does not specify a key management protocol to securely derive and exchange cryptographic keys. RFC4568, *Session Description Protocol (SDP) Security Description for Media Streams*, defines such a protocol specifically designed to exchange cryptographic material using a newly defined SDP *crypto* attribute. Cryptographic parameters are established with only a single message or in single round-trip exchange using the offer/answer model defined in RFC 3264, *An Offer/Answer Model with the Session Description Protocol*.

Release S-C6.2.0 provides support for an initial SDP Security Descriptions (SDES) implementation that generates keys used to encrypt SRTP/SRTCP packets. Authentication of packets will be added to a subsequent release.

A sample SDP exchange is shown below:

The SDP *offerer* sends:

```
v=0
o=sam 2890844526 2890842807 IN IP4 10.47.16.5
s=SRTP Di scussi on
i=A di scussi on of Secure RTP
u=http://www.example.com/semi nars/srtp.pdf
e=marge@example.com (Marge Simpson)
c=IN IP4 168.2.17.12
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto: 1 AES_CM_128_HMAC_SHA1_80
i n i ne: WVNfX19zzW1j dGwgKCKgewkyMj A7fQp9CnVubGVz|2^20|1: 4
```

The SDP *answerer* replies:

```
v=0
o=j i l l 25690844 8070842634 IN IP4 10.47.16.5
s=SRTP Di scussi on
i=A di scussi on of Secure RTP
u=http://www.example.com/semi nars/srtp.pdf
e=homer@example.com (Homer Simpson)
c=IN IP4 168.2.17.11
t=2873397526 2873405696
m=audio 32640 RTP/SAVP 0
a=crypto: 1 AES_CM_128_HMAC_SHA1_80
i n i ne: PS1uQCveeCFCanVmCj kpPywj NWhcYD0mXXtxaVBR|2^20|1: 4
```

The media-level SDP attribute, *crypto*, describes the cryptographic suite, key parameters, and session parameters for the preceding unicast media line. The crypto attribute takes the form:

`a=crypto: tag crypto-sui te key-parameter [session-parameters]`

tag

The tag field contains a decimal number that identifies a specific attribute instance. When an offer contains multiple crypto attributes, the answer uses the tag value to identify the accepted offer.

In the sample offer the tag value is 1.

crypto-suite

The crypto-suite field contains the encryption and authentication algorithms, either AES_CM_128_HMAC_SHA1_80 or AES_CM_128_HMAC_SHA1_32.

The key-parameter field contains one or more sets of keying material for the selected crypto-suite and it has following format.

"inline": " <key||salt> ["|" lifetime] ["|" MKI ":" length]

inline is a method and specifies that the crypto material to be used by the offerer is transmitted via the SDP.

The *key||salt* field contains a base64-encoded concatenated master key and salt.

Assuming the offer is accepted, the key || salt provides the crypto material used by the offerer to encrypt SRTP/SRTCP packets, and used by the answerer to decrypt SRTP/SRTCP packets.

Conversely the key || salt contained in the answer to the offer provides the crypto material used by the answerer to encrypt SRTP/SRTCP packets, and used by the offerer to decrypt SRTP/SRTCP packets.

The *lifetime* field optionally contains the master key lifetime (maximum number of SRTP or SRTCP packets encoded using this master key).

In the sample offer the lifetime value is 1,048, 576 (2^{20}) packets.

The *MKI:length* field optionally contains the Master Key Index (MKI) value and the MKI length.

The MKI is used only when the offer contains multiple keys; it provides a means to differentiate one key from another. The MKI takes the form of an integer, followed by its byte length when included in SRTP/SRTCP packets.

In the sample offer the MKI value is 1 with a length of 4 bytes.

The *session-parameters* field contains a set of optional parameters that may override SRTP session defaults for the SRTP and SRTCP streams.

UNENCRYPTED_SRTP — SRTP messages are not encrypted

UNENCRYPTED_SRTCP — SRTCP messages are not encrypted

UNAUTHENTICATED_SRTP — SRTP messages are not authenticated

When generating an initial offer, the offerer must ensure that there is at least one crypto attribute for each media stream for which security is desired. Each crypto attribute for a given media stream must contain a unique tag. The ordering of multiple crypto attributes is significant — the most preferred crypto suite is listed first.

Upon receiving the initial offer, the answerer must either accept one of the offered crypto attributes, or reject the offer in its entirety.

When an offered crypto attribute is accepted, the crypto attribute in the answer MUST contain the tag and crypto-suite from the accepted crypto attribute in the offer, and the key(s) the answerer will be using for media sent to the offerer.

The crypto-suite is bidirectional and specifies encryption and authentication algorithms for both ends of the connection. The keys are unidirectional in that one key or key set encrypts and decrypts traffic originated by the offerer, while the other key or key set encrypts and decrypts traffic originated by the answerer. The use of symmetric keying, where the same key is used for both encryption and decryption, mandates the key exchange between the offerer and the answerer.

Key exchange via text-based SDP is unacceptable in that malicious network elements could easily eavesdrop and obtain the plaintext keys, thus compromising the privacy and integrity of the encrypted media stream. Consequently, the SDP exchange must be protected by a security protocol such as IPsec or TLS.

Licensing and Hardware Requirements

SRTP/SRTCP support requires the presence of an IPsec NIU and an SSM/SSM2 (Signaling Security Module).

No additional licences are required.

Operational Modes

SRTP topologies can be reduced to three basic topologies which are described in the following sections.

Single-Ended SRTP Termination

Single-ended SRTP termination is illustrated in the following figure.

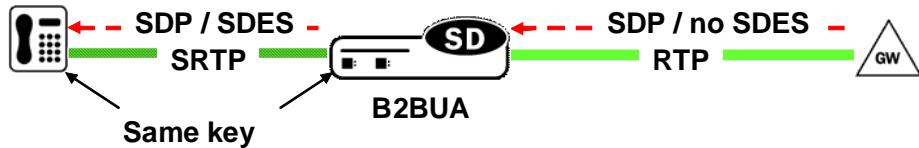


Figure 15 - 1: Single-Ended SRTP Termination

If SRTP is enabled for the inbound realm/interface, the Net-Net SBC handles the incoming call as specified by the Media Security Policy assigned to the inbound realm. If there is crypto attribute contained in the offer, the Net-Net SBC parses the crypto attributes and optional parameters, if any. If the offer contains a crypto attribute or attributes compatible with the requirements specified by the SDES profile assigned to the Media Security policy, it selects the most preferred compatible attribute. Otherwise, the Net-Net SBC rejects the offer. Before the SDP is forwarded to the called party, the Net-Net SBC allocates resources, establishes SRTP and SRTCP Security Associations and updates the SDP by removing the crypto attribute and inserting possibly NAT'ed media addresses and ports. At the same time, the original crypto attribute is also removed from the SDP.

Once the reply from the called party is received, the Net-Net SBC inserts appropriate crypto attribute(s) to form a new SDP, and forward the response back to the calling party.

Refer to *Single-Ended SRTP Termination Configuration* for a sample ACLI configuration.

Back-to-Back SRTP Termination

Back-to-back SRTP termination is illustrated in the following figure.

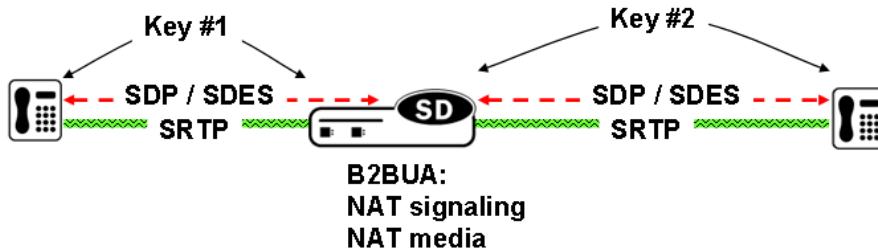


Figure 15 - 2: Back-to-Back SRTP Termination

Initial processing is similar to the single-ended termination described above. Before forwarding the request to the called party, the Net-Net SBC replaces the original crypto attribute with a new one whose crypto attribute conforms to the media security policy for the outbound realm/interface. Upon receiving the answer from the called party, the Net-Net SBC accepts or rejects it, again based upon conformity to the media security policy. If accepted, the Net-Net SBC replaces the original crypto attribute from the called party with its own to form a new SDP, which it forwards back to the calling party. At this point, SRTP media sessions are established on both sides for both calling and called parties.

Refer to *Back-to-Back SRTP Termination Configuration* for a sample ACLI configuration.

SRTP Pass-Thru

SRTP pass-thru is illustrated in the following figure.

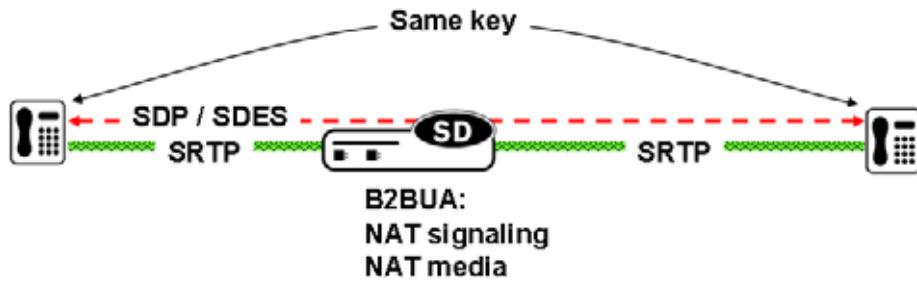


Figure 15 - 3: SRTP Pass-Thru

If the media security policy specifies *pass-through* mode, the Net-Net SBC does not alter the crypto attribute exchange between the calling and the called party; the attribute is transparently passed.

Refer to *SRTP Pass-Thru Configuration* for a sample ACLI configuration.

ACLI Instructions

SDES configuration consists of the following steps.

1. Create one or more SDES profiles which specify parameter values negotiated during the offer/answer exchange.
2. Create one or more Media Security Policies that specify key exchange protocols and protocol-specific profiles.
3. Assign a Media Security Policy to a realm.
4. Create an interface-specific Security Policy (refer to *Security Policy* for a sample ACLI configuration)

SDES Profile Configuration

An SDES profile specifies the parameter values offered or accepted during SDES negotiation.

To configure SDES profile parameters:

- From superuser mode, use the following command sequence to access *sdes-profile* configuration mode.

```
ACMEPACKET# configuration terminal
ACMEPACKET(configuration)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# sdes-profile
ACMEPACKET(sdes-profile)#

```

- Use the required **name** parameter to provide a unique identifier for this *sdes-profile* instance.
name enables the creation of multiple *sdes-profile* instances.
- Use the **crypto-suite** parameter to select the encryption and authentication algorithms accepted or offered by this *sdes-profile*.

Note: SRTP authentication is not currently supported.

Allowable values are:

AES_CM_128_HMAC_SHA1_80 (the default value)

supports AES/128 bit key for encryption and HMAC/SHA-1 80-bit digest for authentication

AES_CM_128_HMAC_SHA1_32

supports AES/128 bit key for encryption and HMAC/SHA-1 32-bit digest for authentication

- Because SRTP authentication is not currently supported, ignore the **srtcp-auth** parameter.
- Use the **srtcp-encrypt** parameter to enable or disable the encryption of RTP packets.

With encryption enabled, the default condition, the Net-Net SBC offers RTP encryption, and rejects an answer that contains an UNENCRYPTED_SRTP session parameter in the crypto attribute.

With encryption disabled, the Net-Net SBC does not offer RTP encryption and includes an UNENCRYPTED_SRTP session parameter in the SDP crypto attribute; it accepts an answer that contains an UNENCRYPTED_SRTP session parameter.

- Use the **srtcp-encrypt** parameter to enable or disable the encryption of RTCP packets.

With encryption enabled, the default condition, the Net-Net SBC offers RTCP encryption, and rejects an answer that contains an UNENCRYPTED_SRTCP session parameter in the crypto attribute.

With encryption disabled, the Net-Net SBC does not offer RTCP encryption and includes an UNENCRYPTED_SRTCP session parameter in the SDP crypto attribute; it accepts an answer that contains an UNENCRYPTED_SRTCP session parameter.

7. Use the **mki** parameter to enable or disable the inclusion of the MKI:length field in the SDP crypto attribute.

The master key identifier (MKI) is an optional field within the SDP crypto attribute that differentiates one key from another. MKI is expressed as a pair of decimal numbers in the form: |mki:mki_length| where mki is the MKI integer value and mki_length is the length of the MKI field in bytes.

The MKI field is necessary only in topologies that may offer multiple keys within the crypto attribute.

Allowable values are enabled and disabled (the default).

enabled – a four-byte MKI field is sent within the crypto attribute

disabled – no MKI field is sent

8. Use **done**, **exit**, and **verify-config** to complete configuration of this SDES profile instance.
9. Repeat Steps 1 through 8 to configure additional SDES profiles.

Media Security Policy Configuration

Use the following procedure to create a Media Security Policy that specifies the role of the Net-Net SBC in the security negotiation. If the SBC takes part in the negotiation, the policy specifies a key exchange protocol and SDES profile for both incoming and outgoing calls.

To configure media-security-policy parameters:

1. From superuser mode, use the following command sequence to access *media-sec-policy* configuration mode.


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# media-sec-policy
ACMEPACKET(media-sec-policy)#

```
2. Use the required **name** parameter to provide a unique identifier for this *media-sec-policy* instance.

name enables the creation of multiple *media-sec-policy* instances.
3. Use optional **pass-thru** parameter to enable or disable pass-thru mode.

With pass-thru mode enabled, the User Agent (UA) endpoints negotiate security parameters between each other; consequently, the Net-Net SBC simply passes SRTP traffic between the two endpoints.

With pass-thru mode disabled (the default state), the Net-Net SBC disallows end-to-end negotiation — rather the Net-Net SBC initiates and terminates SRTP tunnels with both endpoints.
4. Use the **outbound** navigation command to move to *media-sec-outbound* configuration mode. While in this configuration mode you specify security parameters applied to the outbound call leg, that is *calls sent by the Net-Net SBC*.
5. Use the **protocol** parameter to select the key exchange protocol.

Allowable values are *mikey* and *sdes*.

Select *sdes* for SDES key exchange.
6. Use the **profile** parameter to specify the name of the SDES profile applied to calls sent by the Net-Net SBC.

7. Use the **mode** parameter to select the real time transport protocol.
Allowable values are *rtp* and *srtsp* (the default).
mode identifies the transport protocol (RTP or SRTP) included in an SDP offer when this media-security-policy is in effect.
8. Use the **done** and **exit** parameters to return to media-sec-policy configuration mode.
9. Use the **inbound** navigation command to move to *media-sec-inbound* configuration mode. While in this configuration mode you specify security parameters applied to the inbound call leg, that is *calls received by the Net-Net SBC*.
10. Use the **protocol** parameter to select the key exchange protocol.
Allowable values are *mikey* and *sdes*.
Select *sdes* for SDES.
11. Use the **profile** parameter to specify the name of the SDES profile applied to calls received by the Net-Net SBC.
12. Use the **mode** parameter to select the real time transport protocol.
Allowable values are *rtp* and *srtsp* (the default).
mode identifies the transport protocol (RTP or SRTP) accepted in an SDP offer when this media-security-policy is in effect.
13. Use **done**, **exit**, and **verify-config** to complete configuration of this media security policy instance.
14. Repeat Steps 1 through 13 to configure additional media-security policies.

Assign the Media Security Policy to a Realm

To assign a media-security-policy to a realm:

1. From superuser mode, use the following command sequence to access *realm-config* configuration mode. While in this mode, you assign an existing media-security-policy to an existing realm.


```
ACMEPACKET# configure terminal
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(real-m-config)# select
identifier:
  1. access-12
  ...
  ...

selection: 1
ACMEPACKET(real-m-config)#

```
2. Use the **media-sec-policy** parameter to assign the policy to the target realm.
3. Use **done**, **exit**, and **verify-config** to complete assignment of the media-security-policy to the realm.

ACLI Example Configurations

Single-Ended SRTP Termination Configuration

The following section contain relevant sections of system configurations for basic operational modes.

```
ragnarok# show running-configuration

...
...
...

sdes-profile
  name sdes1
  crypto-list AES_CM_128_HMAC_SHA1_80
  srtp-auth enabled
  srtp-encrypt enabled
  srtpc-encrypt enabled
  mki disabled
  key
  salt
  last-modified-by admin@console
  last-modified-date 2009-11-16 15:37:13

media-sec-policy
  name msp2
  pass-through disabled
  inbound
    profile sdes1
    mode srtp
    protocol sdes
  outbound
    profile sdes1
    mode srtp
    protocol sdes
  last-modified-by admin@console
  last-modified-date 2009-11-16 15:37:51

...
...
...

realm-configuration
  identifier peer
  description
  address-prefix 192.168.0.0/16
  network-interfaces M00:0
  mm-in-real-m enabled
  mm-in-network enabled
  mm-same-ip enabled
  mm-in-system enabled
  bw-cac-non-mm disabled
  msm-release disabled
  qos-enabled disabled
  generate-UDP-checksum disabled
  max-bandwidth 0
  fail-back-bandwidth 0
  max-priority-bandwidth 0
  max-latency 0
  max-jitter 0
  max-packet-loss 0
  observ-window-size 0
  parent-real-m
  dns-real-m
  media-policy
  media-sec-policy msp2
```

Back-to-Back SRTP Termination Configuration	in-translation d last-modified-by last-modified-date	admin@console 2009-11-10 15:38:19
	ragnarok# show running-configuration	
	...	
	...	
	...	
	sdes-profile	
	name	sdes1
	crypto-list	AES_CM_128_HMAC_SHA1_80
	srtp-auth	enabled
	srtp-encrypt	enabled
	srtp-encrypt	enabled
	mki	disabled
	key	
	salt	
	last-modified-by	admin@console
	last-modified-date	2009-11-16 15:37:13
	media-security	
	name	msp2
	pass-through	disabled
	inbound	
	profile	sdes1
	mode	srtp
	protocol	sdes
	outbound	
	profile	sdes1
	mode	srtp
	protocol	sdes
	last-modified-by	admin@console
	last-modified-date	2009-11-16 15:37:51
	...	
	...	
	...	
	realm-configuration	
	identifier	peer
	description	192.168.0.0/16
	addr-prefix	M00:0
	network-interfaces	enabled
	mm-in-real-m	enabled
	mm-in-network	enabled
	mm-same-ip	enabled
	mm-in-system	enabled
	bw-cac-non-mm	disabled
	msm-release	disabled
	qos-enabled	disabled
	generate-UDP-checksum	disabled
	max-bandwidth	0
	fallback-bandwidth	0
	max-priority-bandwidth	0
	max-latency	0
	max-jitter	0
	max-packet-loss	0
	observ-window-size	0
	parentrealm	
	dnsrealm	
	mediapolicy	

medi a-sec-pol i cy	msp2
i n-transl ati oni d	
...	
...	
...	
real m-confi g	
i denti fi er	core
descri pti on	
addr-prefi x	172. 16. 0. 0/16
network-i nterfaces	M10: 0
mm-i n-real m	enabl ed
mm-i n-network	enabl ed
mm-Same-i p	enabl ed
mm-i n-system	enabl ed
bw-cac-non-mm	di sabl ed
msm-rel ease	di sabl ed
qos-enabl e	di sabl ed
generate-UDP-checksum	di sabl ed
max-bandwi dth	0
fal l back-bandwi dth	0
max-pri ority-bandwi dth	0
max-l atency	0
max-j itter	0
max-packet-l oss	0
observ-wi ndow-si ze	0
parent-real m	
dns-real m	
medi a-pol i cy	
medi a-sec-pol i cy	msp2
i n-transl ati oni d	
...	
...	
...	
Last-modi fi ed-by	admi n@consol e
Last-modi fi ed-date	2009-11-10 15: 38: 19

SRTP Pass-Thru Configuration

```
ragnarok# show running-configuration
...
...
sdes-profile
  name sdes1
  crypto-list AES_CM_128_HMAC_SHA1_80
  srtp-auth enabled
  srtp-encrypt enabled
  srctp-encrypt enabled
  mki disabled
  key admin
  salt
  last-modified-by admin@console
  last-modified-date 2009-11-16 15:37:13

media-security
  name msp2
  pass-through enabled
  inbound
    profile sdes1
    mode srtp
    protocol sdes
  outbound
    profile sdes1
    mode srtp
    protocol sdes
```

Last-modified-by	admin@console
Last-modified-date	2009-11-16 15:37:51
 ...	
 ...	
 ...	
real m-configuration	
identifier	peer
description	192.168.0.0/16
addr-prefix	M00:0
network-interfaces	enabled
mm-in-real-m	enabled
mm-in-network	enabled
mm-same-ip	enabled
mm-in-system	enabled
bw-cac-non-mm	disabled
msm-relax	disabled
qos-enable	disabled
generate-UDP-checksum	disabled
max-bandwidth	0
fallback-bandwidth	0
max-priority-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observe-window-size	0
parent-real-m	
dns-real-m	
media-policy	msp2
media-sec-policy	
...	
...	
...	
real m-configuration	
identifier	core
description	172.16.0.0/16
addr-prefix	M10:0
network-interfaces	enabled
mm-in-real-m	enabled
mm-in-network	enabled
mm-same-ip	enabled
mm-in-system	enabled
bw-cac-non-mm	disabled
msm-relax	disabled
qos-enable	disabled
generate-UDP-checksum	disabled
max-bandwidth	0
fallback-bandwidth	0
max-priority-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observe-window-size	0
parent-real-m	
dns-real-m	
media-policy	msp2
media-sec-policy	
in-translating	
...	
...	
...	
Last-modified-by	admin@console
Last-modified-date	2009-11-10 15:38:19

Security Policy

A Security Policy enables the Net-Net SBC to identify inbound and outbound media streams that are treated as SRTP/SRTCP. The high-priority Security Policy, *p1*, (shown below) allows signaling traffic from source 172.16.1.3 to destination 172.16.1.10:5060. The lower-priority Security Policy, *p2*, (also shown below) matches media traffic with the same source and destination, but without any specific ports. Consequently, SIP signaling traffic (from local port 5060) go through, but the media stream will be handled by appropriate SRTP SA.

security-policy

name	p1
network-interface	private: 0
priority	0
local-ip-addr-match	172.16.1.3
remote-ip-addr-match	172.16.1.10
local-port-match	5060
remote-port-match	0
trans-protocol-match	UDP
direction	both
local-ip-mask	255.255.255.255
remote-ip-mask	255.255.255.255
action	allow
ike-saiinfo-name	
outbound-sa-filter-name	
local-ip-mask	255.255.255.255
remote-ip-mask	255.255.255.255
local-port-mask	0
remote-port-mask	0
trans-protocol-mask	0
valid	enabled
vlan-mask	0xFFFF
last-modified-by	admin@console
last-modified-date	2009-11-09 15:01:55

security-policy

name	p2
network-interface	private: 0
priority	10
local-ip-addr-match	172.16.1.3
remote-ip-addr-match	172.16.1.10
local-port-match	0
remote-port-match	0
trans-protocol-match	UDP
direction	both
local-ip-mask	255.255.255.255
remote-ip-mask	255.255.255.255
action	srtsp
ike-saiinfo-name	
outbound-sa-filter-name	
local-ip-mask	0.0.0.0
remote-ip-mask	255.255.255.255
local-port-mask	0
remote-port-mask	65535
trans-protocol-mask	255
valid	enabled
vlan-mask	0xFFFF
last-modified-by	admin@console
last-modified-date	2009-11-09 15:38:19

Modified ALCI Configuration Elements

The action parameter in security-policy configuration mode has been modified to accept additional values, *srtcp* and *srtcp*.

- From superuser mode, use the following command sequence to access *media-sec-policy* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)# security-policy
ACMEPACKET(security-policy)# action ?

<enumeration> action (default: ipsec)
ipsec, allow, discard, srtcp, srtcp

ACMEPACKET(security-policy)#

```

Refer to *Security Policy* for sample Security Policies.

The **show security** command has been updated with an **srtcp** option.

```
ACMEPACKET# show security srtcp

sad
spd
statistics
SRTP Statistics
status
```

The **srtcp** option is similar to the **ipsec** option save for the **sad** sub-option that provides data for only SRTP SAs.

The **show sa stats** command has been updated with an **srtcp** option.

```
ACMEPACKET# show sa stats
<ENTER> Show statistics summary of all Security Associations
<ike> Show statistics for IKE Security Associations
<ims-aka> Show statistics for IMS-AKA Security Associations
<srtcp> Show statistics for SRTP Security Associations
sd# show sa stats srtcp
20:06:24-114
```

SA Statistics	----- Lifetime -----		
	Recent	Total	PerMax
SRTP Statistics			
ADD-SA Req Rcvd	0	0	0
ADD-SA Success Resp Sent	0	0	0
ADD-SA Fail Resp Sent	0	0	0
DEL-SA Req Rcvd	0	0	0
DEL-SA Success Resp Sent	0	0	0
DEL-SA Fail Resp Sent	0	0	0
SA Added	0	0	0
SA Add Failed	0	0	0
SA Deleted	0	0	0
SA Delete Failed	0	0	0

Multimedia Internet KEYing Protocol

The Secure Real-Time Transport Protocol, as described in RFC 3711, *The Secure Real-time Transport Protocol (SRTP)*, provides a framework for the encryption and authentication of Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) streams. Both RTP and RTCP are defined by RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*.

Encryption ensures that the call content and associated signalling remains private during transmission. Authentication ensures that (1) received packets are from the purported source, (2) packets are not been tampered with during transmission, and (3) a packet has not been replayed by a malicious server.

A significant number of service providers require the ability to encrypt the content and signalling of their real time communications sessions. There are several approaches to meeting this need, including the use of IPsec encryption as described in 3GPP TS 33.234 I-WLAN, *3G Security: Wireless Local Area Network (WLAN) Interworking Security*. An alternative approach utilizes SRTP as a protocol to encrypt the media, and MIKEY (as defined in RFC 3830, *Multimedia Internet Keying*) to exchange the keying information.

Protocol Overview

While the RFC 3711 framework provides encryption and authentication procedures and defines a set of default cryptographic transforms required for RFC compliance, it does not specify a key management protocol to securely derive and exchange cryptographic keys. RFC3830, *Multimedia Internet Keying*, defines such a protocol that transmits cryptographic material using an exchange of a MIKEY initiation message (referred to as an I_MESSAGE) and a MIKEY response message (referred to as an R_MESSAGE).

Release S-C6.2.0 provides support for an initial MIKEY implementation, based on RFC 3830, and RFC 4567, *Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)*.

The MIKEY I_MESSAGE takes the following format.

HDR, T, RAND, [I Di], [I Dr], {SP}, KEMAC

where

HDR (defined in section 6.1 of RFC 3830) contains the common MIKEY header.

T (defined in section 6.6 of RFC 3830) contains a timestamp.

RAND (defined in section 6.11 of RFC 3830) contains a pseudo-random string used in the generation of TEKs (traffic encryption keys).

I Di and I Dr (defined in section 6.7 of RFC 3830) optionally contain the initiator and responder ID.

SP (defined in sections 6.10 and 6.10.1of RFC 3830) contains a list of security parameters. Pertinent parameters include encryption algorithm, encryption key length, authentication algorithm, authentication key length, salt key length, pseudo random function, key derivation rate, SRTP encryption switch, SRTCP encryption switch, and SRTP authentication switch).

KEMAC (defined in sections 6.2 and 6.13 of RFC 3830) contains TGK (TEK Generation Key) material that generates TEKs.

The MIKEY R_MESSAGE takes the following format.

HDR, T, [IDr], V

where

HDR (defined in section 6.1 of RFC 3830) contains the common MIKEY header.

T (defined in section 6.6 of RFC 3830) contains a timestamp.

[IDr] (defined in section 6.7 of RFC 3830) optionally contains the responder ID.

V (defined in section 6.9 of RFC 3830) contains verification/authentication data.

MIKEY I and R MESSAGES are conveyed in base64-encoded format via SDP using the key management extensions specified in RFC 4567, *Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)*. A sample initiation/response exchange follows.

MIKEY I_MESSAGE

```
v=0
o=alice 2891092738 2891092738 IN IP4 w-I and. example.com
s=Cool stuff
e=alice@w-I and. example.com
t=0 0
c=IN IP4 w-I and. example.com
a=key-mgmt: mikey AQAFgMOXfI ABAAAAAAAAAAAAAsAyONQ6gAAAAAGEE
oo2pee4hp2UaDX8ZE22YwKAAAPZG9uYWxkQGR1Y2suY29tAQAAAAAAQAkOJKpgav
kDaawi9whVBtBtOKZ14ymNuu62+Nv3ozPLygwK/GbAV9i emnGUI Z19fwQUO
SrzKTAv9zV
m=audio 49000 RTP/SAVP 98
a=rtpmap: 98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap: 31 H261/90000
```

The initiator proposes to establish one audio stream and one video stream running SRTP (signaled by the use of the SAVP profile). The initiator offers a session based key to encrypt both audio and video streams by locating the key management *a* line above the *m* lines that describe the audio and video streams. The initiator uses MIKEY to set up the security parameters for SRTP. The MIKEY_I message contains the security parameters, together with the necessary key material.

Upon receiving the offer, the responder validates the MIKEY_I message, and, assuming support for the specified crypto suite and security parameters, accepts the offer and returns a MIKEY_R MESSAGE to the initiator (conveyed via the 200 OK response).

MIKEY R_MESSAGE

```
v=0
o=bob 2891092897 2891092897 IN IP4 foo. example.com
s=Cool stuff
e=bob@foo. example.com
t=0 0
c=IN IP4 foo. example.com
a=key-mgmt: mikeyAQEFgMOXfI ABAAAAAAAAAAAAAYAyONQ6gAAAAAJAA
AQbWIja2V5QG1vdXNI LmNvbQABn8HdGE5BMDXF1uGEga+62AgY5cc=
m=audio 49030 RTP/SAVP 98
a=rtpmap: 98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap: 31 H261/90000
```

Key exchange via text-based SDP is unacceptable in that malicious network elements could easily eavesdrop and obtain the plaintext keys, thus compromising the privacy and integrity of the encrypted media stream. Consequently, the SDP exchange must be protected by a security protocol such as IPsec or TLS.

Licensing and Hardware Requirements

SRTP/SRTCP support requires the presence of an IPsec NIU and an SSM/SSM2 (Signaling Security Module).

No additional licences are required.

Operational Modes

SRTP topologies can be reduced to three basic topologies which are described in the following sections.

Single-Ended SRTP Termination

Single-ended SRTP termination is illustrated in the following figure.

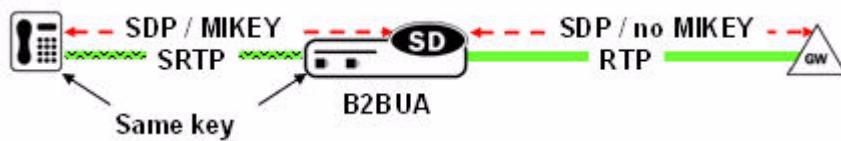


Figure 15 - 4: Single-Ended SRTP Termination

When acting as a responder, upon receiving the SDP offer from a remote caller, the Net-Net SBC confirms that SRTP is enabled on the incoming realm by looking at the inbound mode in the media-sec-policy. If the SRTP is enabled, the SIP process parses the key-mgmt attributes to decrypt the MIKEY I_MESSAGE. After successfully decrypting the message, it accepts or rejects the offer in accordance with the mikey-profile configuration element, assigned to the media-sec-policy. If the Net-Net SBC cannot parse or decrypt the message, the call is rejected.

After obtaining the key and the security parameters from the I_MESSAGE, the Net Net SBC stores these parameters and sends a MIKEY R_MESSAGE to the initiator. It also adds Security Associations to handle the SRTP traffic. If the initiator rejects the response the Net-Net SBC deletes the sessions along with the SRTP/SRTCP SAs.

Acting as an initiator (in the above diagram call coming from gateway to endpoint), the Net-Net SBC confirms that SRTP is enabled on the egress realm by looking at the outbound mode in the media-sec-policy. If SRTP is required, the Net-Net SBC uses MIKEY to generate cryptographic material and parameters (based on the MIKEY profile assigned to the egress realm). After generating cryptographic material, it constructs the MIKEY I_MESSAGE, base64-encodes it, and transmits it via INVITE SDP.

On getting the R_MESSAGE from the call recipient, the Net-Net SBC parses the message and determines if the offer has been accepted. If so, it forwards the answer, without the crypto parameters, back to the call originator. It also adds Security Associations to handle the SRTP traffic. If the initiator rejects the response the Net-Net SBC deletes the sessions along with the SRTP/SRTCP SAs.

Refer to *Single-Ended SRTP Termination Configuration* for a sample ACLI configuration.

Back-to-Back SRTP Termination

Back-to-back SRTP termination is illustrated in the following figure.

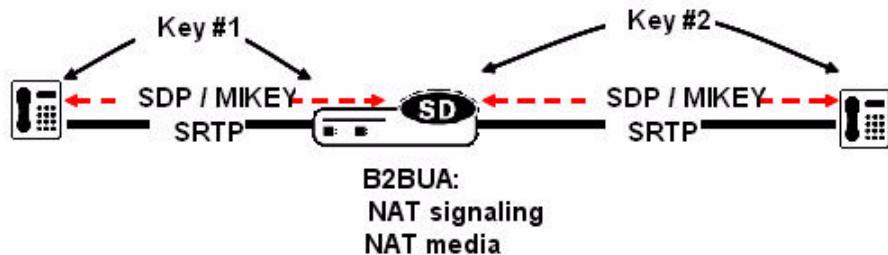


Figure 15 - 5: Back-to-Back SRTP Termination

In this case the Net-Net SBC will act as an initiator on the one side and will act as a responder on the other side. See the description above for the initiator and responder functionality.

Refer to *Back-to-Back SRTP Termination Configuration* for a sample ACLI configuration.

SRTP Pass-Thru

SRTP pass-thru is illustrated in the following figure.

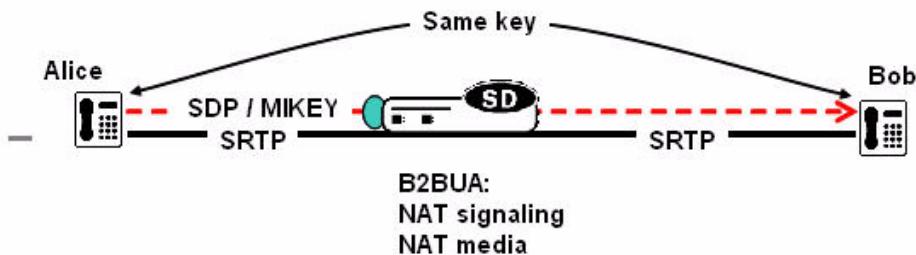


Figure 15 - 6: SRTP Pass-Thru

If the media-sec-policy for the ingress and the egress realms specifies pass-thru mode, the Net-Net SBC does not intercept the SDES/MIKEY exchange between the caller and the callee. The SDES/MIKEY will be forwarded as it is from the caller to the callee, or vice versa.

Refer to *SRTP Pass-Thru Configuration* for a sample ACLI configuration.

ACLI Instructions

MIKEY configuration consists of the following steps.

1. The creation of one or more MIKEY profiles which specify parameter values negotiated during the MIKEY message exchange.
2. The creation of one or more Media Security Policies that specify key exchange protocols and protocol-specific profiles.
3. The assignment of a Media Security Policy to a realm.
4. The creation of an interface-specific Security Policy (refer to *Security Policy* for a sample SCLI configuration)

MIKEY Profile Configuration

A MIKEY profile specifies the parameter values offered as a MILEY initiator or accepted as a MIKEY responder.

To configure MIKEY profile parameters:

- From superuser mode, use the following command sequence to access *mikey-profile* configuration mode.

```
ACMEPACKET# configuration terminal
ACMEPACKET(configuration)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# mikey-profile
ACMEPACKET(mikey-profile)#

```

- Use the required **name** parameter to provide a unique identifier for this *mikey-profile* instance.

name enables the creation of multiple *media-sec-mikey-profile* instances.

- Use the **key-exchange-method** parameter to select the key exchange method that is offered when acting in the initiator role, or accepted when acting in the responder role.

pre-shared (password) is the only currently supported method.

- Use the **encr-algorithm** parameter to select the encryption algorithm that is offered when acting in the initiator role, or accepted when acting in the responder role.

Allowable values are *any*, *NULL*, *AES-CM*

As a MIKEY initiator, select the encryption algorithm to be included in the *I_MESSAGE*.

As a MIKEY responder, select the encryption algorithm to be accepted from the *I_MESSAGE*.

The default value, *any*, indicates that the Net-Net SBC accepts the encryption algorithm specified in the MIKEY_I MESSAGE, as long as it is within the valid range. For the outgoing direction (that is calls originated by the Net-Net SBC), the selection of the encryption algorithm for *any* is as follows:

- If MIKEY is used in the incoming direction the same encryption algorithm is used for the outgoing direction.
- If the incoming direction does not have SRTP, then AES-CM is used as the encryption algorithm.

When *NULL* is used RTP and RTCP packets are not encrypted.

- Use the **auth-algorithm** parameter to select the encryption algorithm that is offered when acting in the initiator role, or accepted when acting in the responder role.

Allowable values are *any*, *NULL*, *HMAC-SHA1-32*, *HMAC-SHA1-80*

As a MIKEY initiator, select the authentication algorithm to be included in the *I_MESSAGE*.

As a MIKEY responder, select the authentication algorithm to be accepted from the *I_MESSAGE*.

The default value, *any*, indicates that the Net-Net SBC accepts the authentication algorithm specified in the MIKEY_I MESSAGE, as long as it is within the valid range. For the outgoing direction (that is calls originated by the Net-Net SBC), the selection of the authentication algorithm for *any* is as follows:

- If MIKEY is used in the incoming direction the same encryption algorithm is used for the outgoing direction.
- If the incoming direction does not have SRTP, then HMAC-SHA1-80 is used as the encryption algorithm.

When NULL is used RTP and RTCP packets are not authenticated.

6. The **shared-secret** parameter contains the shared secret.
A shared secret is not required as MIKEY messages are not protected in this initial implementation.
7. Use the **mki** parameter to enable or disable the inclusion of the MKI:length field in the SDP key-management attribute.
The master key identifier (MKI) is an optional field within the SDP key-management attribute that differentiates one key from another. MKI is expressed as a pair of decimal numbers in the form: |mki:mki_length| where mki is the MKI integer value and mki_length is the length of the MKI field in bytes.
The MKI field is necessary only in topologies that may offer multiple keys within the key-management attribute.
Allowable values are enabled and disabled (the default).
enabled – a four-byte MKI field is sent within the crypto attribute
disabled – no MKI field is sent
8. Use **done**, **exit**, and **verify-config** to complete configuration of this MIKEY profile instance.
9. Repeat Steps 1 through 8 to configure additional MIKEY profiles.

Media Security Policy Configuration

Use the following procedure to create a Media Security Policy that specifies the role of the Net-Net SBC in the security negotiation. If the SBC takes part in the negotiation, the policy specifies a key exchange protocol and SDES profile for both incoming and outgoing calls.

To configure media-security-policy parameters:

1. From superuser mode, use the following command sequence to access *media-sec-policy* configuration mode.


```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# media-security
ACMEPACKET(media-security)# media-sec-policy
ACMEPACKET(media-sec-policy)#

```
2. Use the required **name** parameter to provide a unique identifier for this *media-sec-policy* instance.
name enables the creation of multiple *media-sec-policy* instances.
3. Use optional **pass-thru** parameter to enable or disable pass-thru mode.
With pass-thru mode enabled, the User Agent (UA) endpoints negotiate security parameters between each other; consequently, the Net-Net SBC simply passes SRTP traffic between the two endpoints.
With pass-thru mode disabled, the Net-Net SBC disallows end-to-end negotiation — rather the Net-Net SBC initiates and terminates SRTP tunnels with both endpoints
In the absence of an explicitly configured value, the default value, *disabled*, is used.

4. Use the **outbound** navigation command to move to *media-sec-outbound* configuration mode. While in this configuration mode you specify security parameters applied to the outbound call leg, that is *calls sent by the Net-Net SBC*.
 5. Use the **protocol** parameter to select the key exchange protocol.
Allowable values are *mikey* and *sdes*.
Select *mikey* for MIKEY key exchange.
 6. Use the **profile** parameter to specify the name of the MIKEY profile associated with the outbound leg of this media-security-policy.
 7. Use the **mode** parameter to select the real time transport protocol.
Allowable values are *rtp* and *srtsp* (the default).
mode identifies the transport protocol (RTP or SRTP) included in an SDP offer when this media-security-policy is in effect.
As an initiator, the Net-Net SBC send a MIKEY I_MESSAGE with media descriptions that have the transport protocol specified in this field.
As a responder, the Net-Net SBC accepts a MIKEY I_MESSAGE with media descriptions that have the transport protocol specified in this field. If a media description has no transport protocol that matches the value of this field, the INVITE message is rejected with the 488 Not Acceptable Here response.
 8. Use the **done** and **exit** parameters to return to media-sec-policy configuration mode.
 9. Use the **inbound** navigation command to move to *media-sec-inbound* configuration mode. While in this configuration mode you specify security parameters applied to the inbound call leg, that is calls received by the Net-Net SBC.
 10. Use the **protocol** parameter to select the key exchange protocol.
Allowable values are *mikey* and *sdes*.
Select *mikey* for MIKEY key exchange.
 11. Use the **profile** parameter to specify the name of the MIKEY profile associated with the inbound leg of this media-security-policy.
 12. Use the **mode** parameter to select the real time transport protocol.
Allowable values are *rtp* and *srtsp* (the default).
mode identifies the transport protocol (RTP or SRTP) included in an SDP offer when this media-security-policy is in effect.
As an initiator, the Net-Net SBC sends a MIKEY I_MESSAGE with media descriptions that have the transport protocol specified in this field.
As a responder, the Net-Net SBC accepts a MIKEY I_MESSAGE with media descriptions that have the transport protocol specified in this field. If a media description has no transport protocol that matches the value of this field, the INVITE message is rejected with the 488 Not Acceptable Here response.
 13. Use **done**, **exit**, and **verify-config** to complete configuration of this media security policy instance.
 14. Repeat Steps 1 through 13 to configure additional media-security policies.

Assigning the Media Security Policy to a Realm

To assign a media-security-policy to a realm:

- From superuser mode, use the following command sequence to access *realm-config* configuration mode. While in this mode, you assign an existing media-security-policy to an existing realm.

```
ACMEPACKET# configure terminal
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)# select
identifier:
1. access-10
...
...
selection: 1
ACMEPACKET(realm-config)#

```

- Use the **media-sec-policy** parameter to assign the policy to the current realm.
- Use **done**, **exit**, and **verify-config** to complete assignment of the media-security-policy to the realm.

ACLI Example Configurations

Single-Ended SRTP Termination Configuration

The following section contain relevant sections of system configurations for basic operational modes.

```
ragnarok# show running-config
...
...
mikey-profile
  name mikey1
  key-exchange-method pre-shared
  encr-algorithm any
  auth-algorithm any
  shared-secret
  mki
  last-modified-by admin@console
  last-modified-date 2009-11-15 14:37:13
...
...
mediasec-policy
  name msp2
  pass-through
  inbound
    profile mikey1
    mode srtp
    protocol mikey
  outbound
    profile sdes1
    mode srtp
    protocol mikey
  last-modified-by admin@console
  last-modified-date 2009-11-15 14:37:51
...
...

```

real m-confi g		
identifi er	peer	
descripti on	192.168.0.0/16	
addr-prefix	M00:0	
network-interfaces	enabl ed	
mm-in-real m	enabl ed	
mm-in-network	enabl ed	
mm-same-ip	enabl ed	
mm-in-system	enabl ed	
bw-cac-non-mm	di sabl ed	
msm-rel ease	di sabl ed	
qos-enabl e	di sabl ed	
generate-UDP-checksum	di sabl ed	
max-bandwi dth	0	
fai l back-bandwi dth	0	
max-pri ority-bandwi dth	0	
max-latency	0	
max-jitter	0	
max-packet-loss	0	
observ-wi ndow-si ze	0	
parent-real m		
dns-real m		
medi a-pol i cy		
medi a-sec-pol i cy	msp2	
in-translati on		
...		
...		
...		
last-modifi ed-by	admi n@consol e	
last-modifi ed-date	2009-11-15 14:38:19	

Back-to-Back SRTP Termination Configuration

ragnarok# show runni ng-confi g		
...		
...		
...		
sdes-profil e		
name	sdes1	
crypto-l i st	AES_CM_128_HMAC_SHA1_80	
srtp-auth	enabl ed	
srtp-encrypt	enabl ed	
srtp-encrypt	enabl ed	
mki	di sabl ed	
key		
salt		
last-modifi ed-by	admi n@consol e	
last-modifi ed-date	2009-11-16 15:37:13	
medi a-sec-pol i cy		
name	msp2	
pass-through	di sabl ed	
inbound		
profil e	sdes1	
mode	srtp	
protocol	sdes	
outbound		
profil e	sdes1	
mode	srtp	
protocol	sdes	
last-modifi ed-by	admi n@consol e	
last-modifi ed-date	2009-11-16 15:37:51	
...		
...		
...		

real m-config	
identifer	peer
description	192.168.0.0/16
addr-prefix	M00:0
network-interfaces	enabled
mm-in-real m	enabled
mm-in-network	enabled
mm-same-ip	enabled
mm-in-system	enabled
bw-cac-non-mm	disabled
msm-release	disabled
qos-enable	disabled
generate-UDP-checksum	disabled
max-bandwidth	0
fallback-bandwidth	0
max-priority-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observ-window-size	0
parent-real m	
dns-real m	
media-pol icy	msp2
media-sec-pol icy	
in-translations	
...	
...	
...	
real m-config	
identifer	core
description	172.16.0.0/16
addr-prefix	M10:0
network-interfaces	enabled
mm-in-real m	enabled
mm-in-network	enabled
mm-same-ip	enabled
mm-in-system	enabled
bw-cac-non-mm	disabled
msm-release	disabled
qos-enable	disabled
generate-UDP-checksum	disabled
max-bandwidth	0
fallback-bandwidth	0
max-priority-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observ-window-size	0
parent-real m	
dns-real m	
media-pol icy	msp2
media-sec-pol icy	
in-translations	
...	
...	
...	
last-modified-by	admin@console
last-modified-date	2009-11-10 15:38:19

SRTP Pass-Thru Configuration

```
ragnarok# show running-configuration

...
...
...

sdes-profile
  name sdes1
  crypto-list AES_CM_128_HMAC_SHA1_80
  srtp-auth enabled
  srtp-encrypt enabled
  srctp-encrypt enabled
  mki disabled
  key
  salt
  last-modified-by admn@console
  last-modified-date 2009-11-16 15:37:13
  media-security
    name msp2
    pass-through inbound
      profile sdes1
      mode srtp
      protocol sdes
    outbound
      profile sdes1
      mode srtp
      protocol sdes
  last-modified-by admn@console
  last-modified-date 2009-11-16 15:37:51
  ...
  ...
  ...

real-m-configuration
  identifier peer
  description 192.168.0.0/16
  address-prefix M00:0
  network-interfaces enabled
  mm-in-real-m enabled
  mm-in-network enabled
  mm-same-ip enabled
  mm-in-system enabled
  bw-cac-non-mm disabled
  msm-release disabled
  qos-enabled disabled
  generate-UDP-checksum disabled
  max-bandwidth 0
  fail-back-bandwidth 0
  max-priority-bandwidth 0
  max-latency 0
  max-jitter 0
  max-packet-loss 0
  observ-window-size 0
  parentrealm
  dnsrealm
  mediapolicy
  mediasecpolicy
  ...
  ...
  ...

real-m-configuration
  identifier core
  description 172.16.0.0/16
  address-prefix M10:0
  network-interfaces enabled
  mm-in-real-m enabled
  mm-in-network enabled
```

mm-same-ip	enabled
mm-in-system	enabled
bw-cac-non-mm	disabled
msm-relax	disabled
qos-enabled	disabled
generate-UDP-checksum	disabled
max-bandwidth	0
fallback-bandwidth	0
max-primary-bandwidth	0
max-latency	0
max-jitter	0
max-packet-loss	0
observe-window-size	0
parent-real-m	
dns-real-m	
media-policy	
media-sec-policy	msp2
in-translating	
...	
...	
Last-modified-by	admin@console
Last-modified-date	2009-11-10 15:38:19

Security Policy

A Security Policy enables the Net-Net SBC to identify inbound and outbound media streams that are treated as SRTP/SRTCP. The high-priority Security Policy, *p1*, (shown below) allows signaling traffic from source 172.16.1.3 to destination 172.16.1.10:5060. The lower-priority Security Policy, *p2*, (also shown below) matches media traffic with the same source and destination, but without any specific ports. Consequently, SIP signaling traffic (from local port 5060) go through, but the media stream will be handled by appropriate SRTP SA.

security-policy	
name	p1
network-interface	private: 0
priority	0
local-ip-addr-match	172.16.1.3
remote-ip-addr-match	172.16.1.10
local-port-match	5060
remote-port-match	0
trans-protocol-match	UDP
direction	both
local-ip-mask	255.255.255.255
remote-ip-mask	255.255.255.255
action	allow
ike-sai-name	
outbound-sa-fine-grained-mask	
local-ip-mask	255.255.255.255
remote-ip-mask	255.255.255.255
local-port-mask	0
remote-port-mask	0
trans-protocol-mask	0
valid	enabled
vlan-mask	0xFFFF
Last-modified-by	admin@console
Last-modified-date	2009-11-09 15:01:55

```

security-policy
  name p2
  network-interface private:0
  priority 10
  local-ip-addr-match 172.16.1.3
  remote-ip-addr-match 172.16.1.10
  local-port-match 0
  remote-port-match 0
  trans-protocol-match UDP
  direction both
  local-ip-mask 255.255.255.255
  remote-ip-mask 255.255.255.255
  action srtp
  key-sai info-name
  outbound-sa-fine-grained-mask
    local-ip-mask 0.0.0.0
    remote-ip-mask 255.255.255.255
    local-port-mask 0
    remote-port-mask 65535
    trans-protocol-mask 255
    valid enabled
    vlan-mask 0xFFFF
  last-modified-by admin@console
  last-modified-date 2009-11-09 15:38:19

```

Modified ALCI Configuration Elements

The action parameter in security-policy configuration mode has been modified to accept additional values, *srtcp* and *srtcp*.

- From superuser mode, use the following command sequence to access *media-sec-policy* configuration mode.

```

ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)# security-policy?
ACMEPACKET(security-policy)# action?

<enumeration> action (default: ipsec)
      ipsec, allow, discard, srtp, srtcp

ACMEPACKET(security-policy)#

```

Refer to *Security Policy* for sample Security Policies.

The **show sa stats** command has been enhanced.

```

ACMEPACKET# show sa stats srtp
12:27:45-153
SA Statistics          ----- Lifetime -----
                                         Recent   Total   PerMax
SRTP Statistics
ADD-SA Req Rcvd        0        4        4
ADD-SA Success Resp Sent 0        3        3
ADD-SA Fail Resp Sent  0        1        1
DEL-SA Req Rcvd        0        0        0

```

DEL-SA Success Resp Sent	0	0	0
DEL-SA Fail Resp Sent	0	0	0
SA Added	0	3	3
SA Add Failed	0	1	1
SA Deleted	0	0	0
SA Delete Failed	0	0	0

ACMEPACKET# show sa stats

12: 36: 00-148

	----- Lifetime -----		
	Recent	Total	PerMax
IKE Statistics			
ADD-SA Req Rcvd	0	4	4
ADD-SA Success Resp Sent	0	3	3
ADD-SA Fail Resp Sent	0	1	1
DEL-SA Req Rcvd	0	0	0
DEL-SA Success Resp Sent	0	0	0
DEL-SA Fail Resp Sent	0	0	0
ACQUIRE-SA Req Sent	0	5	5
ACQUIRE-SA Success Resp	0	5	5
ACQUIRE-SA Fail Resp Rcv	0	0	0
ACQUIRE-SA Trans Timeout	0	0	0
SA Added	0	3	3
SA Add Failed	0	1	1
SA Deleted	0	0	0
SA Delete Failed	0	0	0
IMS-AKA Statistics			
ADD-SA Req Rcvd	0	0	0
ADD-SA Success Resp Sent	0	0	0
ADD-SA Fail Resp Sent	0	0	0
DEL-SA Req Rcvd	0	0	0
DEL-SA Success Resp Sent	0	0	0
DEL-SA Fail Resp Sent	0	0	0
SA Added	0	0	0
SA Add Failed	0	0	0
SA Deleted	0	0	0
SA Delete Failed	0	0	0
SRTP Statistics			
ADD-SA Req Rcvd	0	0	0
ADD-SA Success Resp Sent	0	0	0
ADD-SA Fail Resp Sent	0	0	0
DEL-SA Req Rcvd	0	0	0
DEL-SA Success Resp Sent	0	0	0
DEL-SA Fail Resp Sent	0	0	0
SA Added	0	0	0
SA Add Failed	0	0	0
SA Deleted	0	0	0
SA Delete Failed	0	0	0

ACMEPACKET#

IPSec Support

The Net-Net SBC offers hardware-based IPSec for securing signaling, media, and management traffic at the network layer. This feature is supported by a 2-port copper (10/100/1000) or a 2-port SFP GigE Optical IPSec accelerated physical interface card on the Net-Net SBC.

In Net-Net OS Release C5.0, 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:

- DES
- 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.

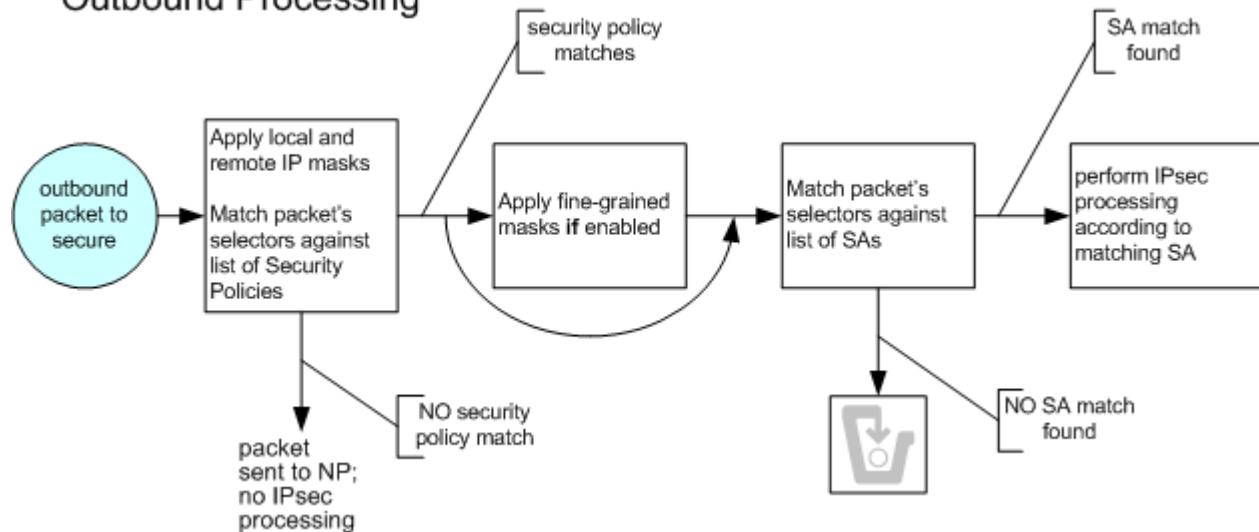
IPSec Implementation

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.

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
- Destination IP subnet mask
- VLAN mask

By default, the fine-grained security policy is set to match and pass all traffic untouched to the security association (SA) portion of IPSec processing.

Fine-grained policy selection works by performing a logical AND between outbound traffic's fine-grained selectors and the traffic's corresponding attributes. The result is then used to find the matching SA. Applying a fine-grained mask has the effect of forcing a contiguous block of IP addresses and/or ports to appear as one address and or port. During the next step of IPSec processing, when an SA is chosen, the 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 optionally applies fine-grained masking, an SA lookup is performed on the traffic. An SA is the set of rules that define the association between two endpoints or entities that create the secured communication. To choose an SA, the Net-Net SBC searches for a match against the outgoing traffic's SA selectors. SA selectors are as follows:

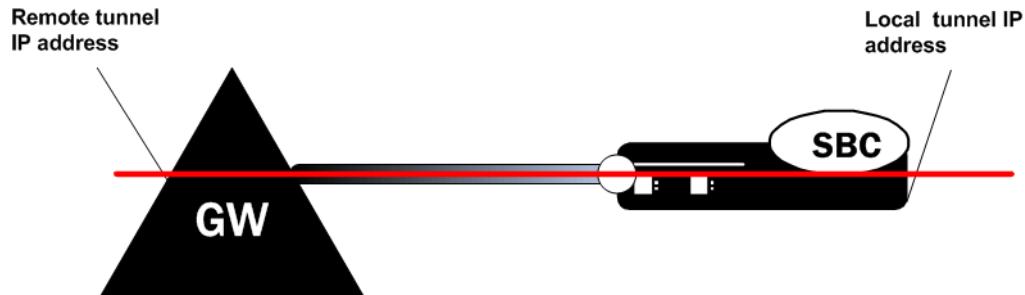
- VLAN
- Source IP address
- Source IP port
- Destination IP address
- Destination IP port
- Transport Protocol

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.

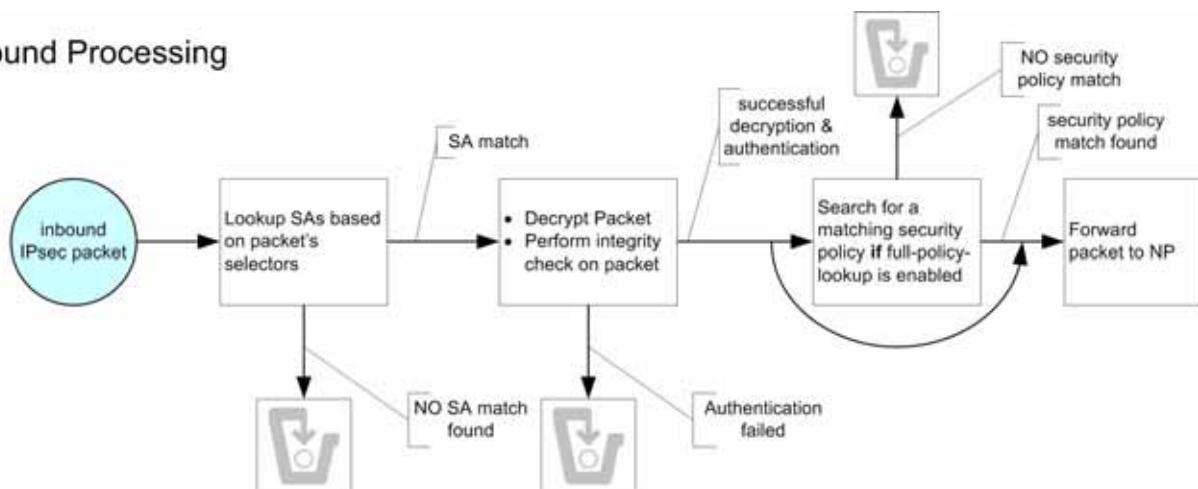


The authentication algorithm and the authentication key must always be configured. The Net-Net SBC supports hmac-md5 or hmac-sha1 authentication algorithms. Because only manual keying is supported, the key must be entered by hand. When encryption is required, the encryption algorithm and the encryption key must be configured. The Net-Net SBC supports des, 3des, aes-128-cbc, aes-256-cbc, aes-128-ctr, and aes-256-ctr encryption algorithms. When using the two encryption protocols that operate in AES counter mode (RFC 3686), an additional nonce value is required. In addition, the security parameter index (SPI) must be configured for each SA. All SPI values 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)
- Source IP address (plus mask)
- Source IP port
- Destination IP address (plus mask)
- Destination IP port
- Transport Protocol
- SPI

If no matching SA is found, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG. When the 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.

Inbound Full Policy Lookup

Inbound traffic can optionally be subjected to a full policy lookup, after decryption and authentication. A full policy lookup checks if a security policy exists for this inbound traffic before the Net-Net SBC sends the decrypted packet to the NP. If no matching security policy is found, even after a successful SA match, the packets are discarded, and optionally dumped to secured.log if the log-level is set to DEBUG.

Full policy lookups are useful for traffic filtering. If you wish to drop traffic not sent to a specific port (e.g. drop any traffic not sent to port 5060), a security policy with specific remote-port-match parameter would be used to define what is valid (i.e., not dropped). Full policy lookups may degrade system performance because they consume additional processing; they should not be configured unless absolutely useful.

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.

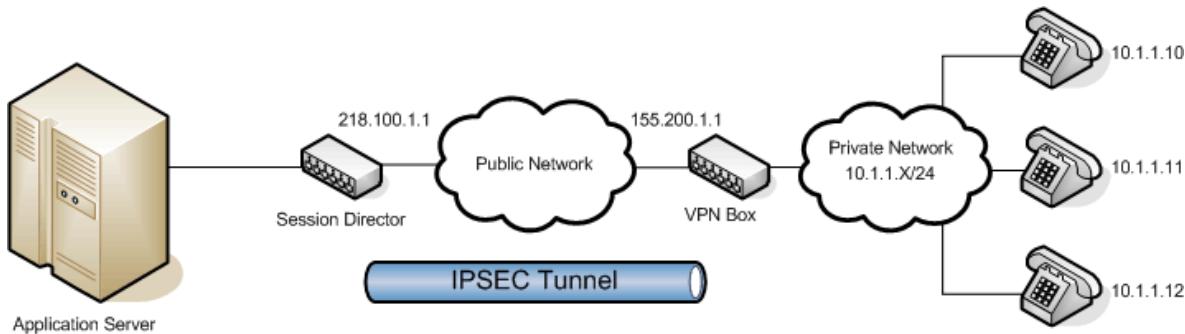
When a failover occurs between HA peers, the newly-active system starts sending traffic with the IPSec sequence number starting at 0. A remote system's anti-replay mechanism observes this and labels the traffic as insecure. It is therefore recommended that anti-replay protection not be used with Net-Net 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 as well as the inbound SA mask allows any number of endpoints on the 10.1.1.X network to use a single security association (a coarse-grain configuration). The configuration in this example follows:

A packet sent from the 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 used, three discrete SAs would have to be configured: one for each of the three phones.

```
ACMEPACKET(manual) # show
manual
  name          assoc1
  spi           1516
  network-interface lefty: 0
  local-ip-addr 100.20.50.7
  remote-ip-addr 100.25.56.10
  local-port    60035
  remote-port   26555
  trans-protocol ALL
  ipsec-protocol esp
  direction     both
  ipsec-mode    tunnel
  auth-algo     hmac-md5
  encr-algo    des
  auth-key
```

encr-key		
aes-ctr-nonce		0
tunnel-mode		
local-ip-addr	100.20.55.1	
remote-ip-addr	101.22.54.3	
last-modified-date	2007-04-30 16:04:46	

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# config terminal
ACMEPACKET(config)#

```
2. Type **security** and press <Enter> to access the security path of the configuration menu.

```
ACMEPACKET(config)# security
ACMEPACKET(security)#

```
3. Type **ipsec** and press <Enter>.

```
ACMEPACKET(security)# ipsec
ACMEPACKET(ipsec)#

```
4. Type **security-policy** and press <Enter>. The prompt changes to let you know that you can begin configuration.

```
ACMEPACKET(ipsec)# security-policy
ACMEPACKET(security-policy)#

```
5. **name**—Enter a name for this security policy. This parameter is required and has no default.
6. **network-interface**—Enter the network interface and VLAN where this security policy applies in the form: interface-name:VLAN
7. **priority**—Enter the priority number of this security policy. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—254
8. **action**—Enter the action the Net-Net SBC should take when this policy matches outbound IPSec traffic. The default value is **ipsec**. The valid values are:
 - **ipsec**—Continue processing as IPSec traffic
 - **allow**—Forward the traffic without any security processing
 - **discard**—Discard the traffic
9. **direction**—Enter the direction of traffic this security policy can apply to. The default value is **both**. The valid values are:
 - **in**—This security policy is valid for inbound traffic

- **out**—This security policy is valid for outbound traffic
 - **both**—This security policy is valid for inbound and outbound traffic
- To define the criteria for matching traffic selectors for this security policy:
10. **local-ip-addr-match**—Enter the source IP address to match. The default value is **0.0.0.0**.
 11. **remote-ip-addr-match**—Enter the destination IP address to match. The default value is **0.0.0.0**.
 12. **local-port-match**—Enter the source port to match. A value of 0 disables this selector. The default value is zero (**0**). The valid range is:
 - Minimum—0
 - Maximum—65535
 13. **remote-port-match**—Enter the destination port to match. A value of 0 disables this selector. The default value is zero (**0**). The valid range is:
 - Minimum—0
 - Maximum—65535
 14. **trans-protocol-match**—Enter the transport protocol to match. The default value is **all**. The valid values are:
 - UDP | TCP | ICMP | ALL
 15. **local-ip-mask**—Enter the source IP address mask, in dotted-decimal notation. The default value is **255.255.255.255**.
 16. **remote-ip-mask**—Enter the remote IP address mask, in dotted-decimal notation. The default value is **255.255.255.255**.
 17. Save your work using the ACLI **done** command.

Defining Outbound Fine-Grained SA Matching Criteria

To define outbound fine-grained SA matching criteria:

1. From within the security policy configuration, type **outbound-sa-fine-grained-mask** and press <Enter>. The prompt changes to let you know that you can begin configuration.
2. **local-ip-mask**—Enter the fine-grained source IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
3. **remote-ip-mask**—Enter the fine-grained destination IP address mask to apply to outbound IP packets for SA matching. Valid values are in dotted-decimal notation. The default mask matches for all traffic.
4. **local-port-mask**—Enter the local port mask for this security policy. The default value for this parameter is **0**. The valid range is:
 - Minimum—0
 - Maximum—65535
5. **remote-port-mask**—Enter the remote port mask for this security policy. The default value for this parameter is **0**. The valid range is:
 - Minimum—0
 - Maximum—65535
6. **trans-protocol-mask**—Enter the transport protocol mask for this security policy. The default value for this parameter is **0**. The valid range is:

- Minimum—0
 - Maximum—255
7. **vlan-mask**—Enter the fine-grained VLAN mask to apply to outbound IP packets for SA matching. The default is **0x000 (disabled)**. The valid range is:
 - 0x000 - 0xFFFF
 8. Save your work using the ACLI **done** command.

Configuring an IPSec SA**To configure an IPSec SA:**

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **configure terminal**
 ACMEPACKET(configure)#
2. Type **security** and press <Enter> to access the security path of the configuration menu.
 ACMEPACKET(configure)# **security**
 ACMEPACKET(security)#
3. Type **ipsec** and press <Enter>.
 ACMEPACKET(security)# **ipsec**
 ACMEPACKET(ipsec)#
4. Type **security-association** and press <Enter>.
 ACMEPACKET(ipsec)# **security-association**
 ACMEPACKET(security-association)#
5. Type **manual** and press <Enter>. The prompt changes to let you know that you can begin configuration.
 ACMEPACKET(security-association)# **manual**
 ACMEPACKET(manual)#
6. **name**—Enter a name for this security policy.
7. **network-interface**—Enter the network interface and VLAN where this security policy applies in the form: **interface_name:VLAN**
8. **direction**—Enter the direction of traffic this security policy can apply to. The default value is **both**. Valid values are:
 - in | out | both
9. Save your work using the ACLI **done** command.

Defining Criteria for Matching Traffic Selectors per SA**To define the criteria for matching traffic selectors for this SA:**

1. From within the **manual** portion of the security association configuration, you need to set the parameters described in this process.
 ACMEPACKET(security-association)# **manual**
 ACMEPACKET(manual)#
2. **local-ip-addr**—Enter the source IP address to match.
3. **remote-ip-addr**—Enter the destination IP address to match.
4. **local-port**—Enter the source port to match. A value of **0** disables this selector. The default value is **0**, disabling this parameter. The valid range is:
 - Minimum—0

- Maximum—65535
5. **remote-port**—Enter the destination port to match. A value of **0** disables this selector. The default value is **0**, disabling this parameter. The valid range is:
 - Minimum—0
 - Maximum—65535
 6. **trans-protocol**—Enter the transport protocol to match for traffic selectors for this SA. The default value is **ALL**. The valid values are:
 - UDP | TCP | ICMP | ALL
 7. **ipsec-protocol**—Select the IPSec protocol to use for this SA configuration. The default value for this parameter is **esp**. Valid values are:
 - esp | ah
 8. **spi**—Enter the security parameter index. The default value is **256**. The valid range is:
 - Minimum—256
 - Maximum—2302
 9. **ipsec-mode**—Enter the IPSec mode of this SA. The default value is **transport**. The valid values are:
 - tunnel | transport
 10. **auth-algo**—Enter the IPSec authentication algorithm for this SA. The default value is **null**. The valid values are:
 - hmac-md5 | hmac-sha1 | null
 11. **auth-key**—Enter the authentication key for the previously chosen authentication algorithm for this SA.
- Note:** The specified **auth-key** value will be encrypted in the configuration and will no longer be visible in clear-text.
12. **encr-algo**—Enter the IPSec encryption algorithm for this SA. The default value is **null**. The valid values are:
 - des | 3des | aes-128-cbc | aes-256-cbc | aes-128-ctr | aes-256-ctr | null
 13. **encr-key**—Enter the encryption key for the previously chosen encryption algorithm for this SA.
- Note:** The specified **encr-key** value will be encrypted in the configuration and will no longer be visible in clear-text.
14. **aes-ctr-nonce**—Enter the AES nonce if aes-128-ctr or aes-256-ctr were chosen as your encryption algorithm. The default value is **0**.

Defining Endpoints for IPSec Tunnel Mode

To define endpoints for IPSec tunnel mode:

1. From within the **manual** portion of the security association configuration, you need to set the parameters described in this process.

```
ACMEPACKET(security-association) # manual
ACMEPACKET(manual) #
```
2. **local-ip-addr**—Enter the local public IP address which terminates the IPSec tunnel.

3. **remote-ip-addr**—Enter the remote public IP address which terminates the IPSec tunnel.
4. Save your work using the ACLI **done** command.

Real-Time IPSec Process Control

The **notify secured** commands force the IPSec application to perform tasks in real-time, outside of the Net-Net SBC reloading and activating the running configuration. The **notify secured** usage is as follows:

```
notify secured [activateconfig | nolog | log | debug | nodebug]
```

The following arguments perform the listed tasks:

- **nolog**—Disables secured logging
- **log**—Enables secured logging
- **debug**—Sets secured log level to DEBUG
- **nodebug**—Sets secured log level to INFO

Key Generation

The **generate-key** command generates keys for the supported encryption or authentication algorithms supported by the 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:

```
generate-key [hmac-md5 | hmac-sha1 | aes-128 | aes-256 | des | 3des]
```

IDS Reporting

The Net-Net SBC supports a wide range of intrusion detection and protection capabilities for vulnerability and attack profiles identified to date. The IDS reporting feature is useful for enterprise customers requirement to report on intrusions and suspicious behavior that it currently monitors.

IDS Licensing

This feature requires the IDS Reporting license. Note the following capabilities and restrictions of the license:

- The following configuration parameters located in the **access control** and **media manager configuration** elements are only visible after installing the license:
 - trap-on-demote-to-deny
 - syslog-on-demote-to-deny
 - cac-failure-threshold
 - untrust-cac-failure-threshold
- Endpoint demotions based on admission control failures are only a valid option with the IDS License.
- The presence of the IDS license makes the `apSysMgmtInetAddrWithReasonDOSTrap` trap available and the

apSysMgmtExpDOSTrap unavailable. WIthout an IDS licence installed, only the apSysMgmtExpDOSTrap trap is available.

- The **Trust->Untrust** and **Untrust-Deny** counters in the SIP and MGCP ACLs' statistics are visible regardless of the IDS license's presence.
- The **Demote Trust-Untrust** and **Demote Untrust-Deny** collect records in the SIP and MGCP ACL HDR groups are visible regardless of the IDS license's presence.
- A GET operation can be preformed on the two MIB entries to view the global endpoint counter for Demotions from Trusted to Untrusted and from Untrusted to Deny regardless of the IDS license's presence
- On Net-Net 3800 systems, the DOS license must be installed in addition to the IDS license in order to enable all features described in this section.

Basic Endpoint Demotion Behavior

Each session agent or endpoint is promoted or demoted among the trusted, untrusted, and denied queues depending on the **trust-level** parameter of the session agent or realm to which it belongs. Users can also configure access control rules to further classify signaling traffic so it can be promoted or demoted among trust queues as necessary.

An endpoint can be demoted in two cases:

1. Net-Net SBC receiving too many signaling packets within the configured time window (**maximum signal threshold** in **realm config** or **access control**)
2. Net-Net SBC receiving too many invalid signaling packets within the configured time window. (**invalid signal threshold** in **realm config** or **access control**)

Endpoint Demotion Reporting

The Net-Net SBC counts the number of endpoint or session agent promotions and demotions. Further, the Net-Net SBC counts when endpoints or session agents transition from trusted to untrusted and when endpoints transition from untrusted to denied queues. These counts are maintained for SIP and MGCP signaling applications. They appear as the Trust->Untrust and Untrust->Deny rows in the show sipd acls (998) and show mgcp acls (999) commands.

SNMP Reporting

These per-endpoint counters are available under APSYSMGMT-MIB -> acmepacketMgmt -> apSystemManagementModule -> apSysMgmtMIBObjects -> apSysMgmtMIBGeneralObjects.

MIB NAME	MIB OID	PURPOSE
apSysSipEndptDemTrustToUntrust	.1.3.6.1.4.1.9148.3.2.1.1.19	Global counter for SIP endpoint demotions from trusted to untrusted.
apSysSipEndptDemUntrustToDeny	.1.3.6.1.4.1.9148.3.2.1.1.20	Global counter for SIP endpoint demotions from untrusted to denied.

MIB NAME	MIB OID	PURPOSE
apSysMgcpEndptDemTrustToUntrust	.1.3.6.1.4.1.9148.3.2.1.1.21	Global counter for MGCP endpoint demotions from trusted to untrusted.
apSysMgcpEndptDemUntrustToDeny	.1.3.6.1.4.1.9148.3.2.1.1.22	Global counter for MGCP endpoint demotions from untrusted to denied.

HDR Reporting

The SIP (sip-ACL-oper) and MGCP (mgcp-oper) HDR ACL status collection groups include the following two metrics:

- Demote Trust-Untrust (Global counter of endpoint demotion from trusted to untrusted queue)
- Demote Untrust-Deny (Global counter of endpoint demotion from untrusted to denied queue)

Endpoint Demotion SNMP Traps

An SNMP trap can be sent when the Net-Net SBC demotes an endpoint to the denied queue. This is set by enabling the **trap on demote to deny** parameter located in the **media manager config** configuration element.

When the IDS license is installed and the **trap on demote to deny** parameter is enabled, apSysMgmtInetAddrWithReasonDOSTrap trap is sent. This trap supersedes the apSysMgmtInetAddrDOSTrap trap.

When the IDS license is installed and the **trap on demote to deny** parameter is disabled the apSysMgmtInetAddrWithReasonDOSTrap trap is not sent from the Net-Net SBC, even when an endpoint is demoted to the denied queue.

This apSysMgmtInetAddrWithReasonDOSTrap contains the following data:

- apSysMgmtDOSInetAddressType—Blocked IP address family (IPv4 or IPv6)
- apSysMgmtDOSInetAddress—Blocked IP address
- apSysMgmtDOSRealmID—Blocked Realm ID
- apSysMgmtDOSFromURI—The FROM header of the message that caused the block (If available)
- apSysMgmtDOSReason—The reason for demoting the endpoint to the denied queue: This field can report the following three values:
 - Too many errors
 - Too many messages
 - Too many admission control failures

Note: By default, this parameter is enabled for upgrade configurations, as the current behavior is to send a trap for every endpoint that is demoted to deny. However, for a new configuration created, the value to this configuration is disabled.

Endpoint Demotion Syslog Message

A Syslog message can be generated when an endpoint is demoted. Setting the **media manager config -> syslog-on-demote-to-deny** parameter to enabled writes an endpoint demotion warning to the syslog every time an endpoint is

demoted to the denied queue. By default, this configuration option is set to disabled. The syslog message has a WARNING Level and looks like this:

```
Jan 15 12:22:48 172.30.60.12 ACMESYSTEM si pd[1c6e0b90] WARNING
SigAddr[access: 168.192.24.40; 0=allow: DENY] ttl=3632 guard=798 exp=30
Demoted to Black-List (Too many admission control failures)
```

ACL Configuration and Examples

To configure the Net-Net SBC to send traps and/or write syslog messages on endpoint demotion:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **configure terminal**
2. Type **media-manager** and press <Enter> to access the media-level configuration elements.
ACMEPACKET(config)# **media-manager**
ACMEPACKET(media-manager)#
3. Type **media-manager** and press <Enter>.
ACMEPACKET(media-manager)# **media-manager**
ACMEPACKET(media-manager-config)#
4. **trap-on-demote-to-deny**—Set this parameter to enabled for the Net-Net SBC to send the apSysMgmtNetAddrWithReasonDOSTrap trap when applicable.
5. **syslog-on-demote-to-deny**—Set this parameter to enabled for the Net-Net SBC to write an endpoint demotion warning message to the syslog.
6. Save your work.

Endpoint Demotion due to CAC coverage

The Net-Net SBC can demote endpoints from trusted to untrusted queues when CAC failures exceed a configured threshold. The Net-Net SBC can also demote endpoints from untrusted to denied queues when CAC failures exceed another configured threshold.

The Net-Net SBC maintains CAC failures per-endpoint. The CAC failure counter is incremented upon certain admission control failures only if either one of **cac-failure-threshold** or **untrust-cac-fail-threshold** is non-zero.

The **cac failure threshold** parameter is available in the access control and realm configuration elements. Exceeding this parameter demotes an endpoint from the trusted queue to the untrusted queue. The **untrust cac-failure-threshold** parameter is available in the access control and realm configuration elements. Exceeding this parameter demotes an endpoint from the untrusted queue to the denied queue.

If both the **cac failure threshold** and **untrusted cac failure threshold** are configured to 0, then admission control failures are considered and counted as invalid signaling messages for determining if the **invalid-signal-threshold** parameter value has been exceeded.

CAC Attributes used for Endpoint Demotion

The Net-Net SBC determines CAC failures only by considering the *calling* endpoint's signaling messages traversing the *calling realms' configuration*. If an endpoint exceeds the following CAC thresholds, the Net-Net SBC will demote the endpoint when the CAC failure thresholds are enabled.

1. **sip-interface user CAC sessions (realm-config > user-cac-sessions)**

2. sip-interface user CAC bandwidth (realm-config > user-cac-bandwidth)
3. External policy server rejects a session

Authentication Failures used for Endpoint Demotion

If an endpoint fails to authenticate with the Net-Net SBC using SIP HTTP digest authentication OR endpoint fails authentication with an INVITE with authentication incase registration-caching is disabled, and receives back a 401 or 407 response from the registrar

When the Net-Net SBC receives a 401 or 407 message from the registrar in response to one of the following conditions, the endpoint attempting authentication is demoted.

- endpoint fails to authenticate with the Net-Net SBC using SIP HTTP digest authentication
- endpoint fails to authenticate with the Net-Net SBC using INVITE message when registration-caching is disabled

ACLI Configuration and Examples

To configure endpoint demotion on CAC failures:

1. In Superuser mode, type **configure terminal** and press <Enter>.
 ACMEPACKET# **configure terminal**
 ACMEPACKET(configure)#
2. Type **session-router** and press <Enter>.
 ACMEPACKET(configure)# **session-router**
 ACMEPACKET(session-router)#
3. Type **access-control** and press <Enter>.
 ACMEPACKET(session-router)# **access-control**
 ACMEPACKET(access-control)#

If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.
4. **cac-failure-threshold**—Enter the number of CAC failures for any single endpoint that will demote it from the trusted queue to the untrusted queue.
5. **untrust-cac-failure-threshold**—Enter the number of CAC failures for any single endpoint that will demote it from the untrusted queue to the denied queue.
6. Save your work.

Maintenance and Troubleshooting

show sipd acls

The **show sipd acls** command includes counters that track the number of endpoints demoted from trusted to untrusted and the number of endpoints demoted from untrusted to denied. For example:

```
ACMEPACKET# show sipd acls
16:49:07-133
SIP ACL Status          -- Period -- ----- Lifetime -----
                           Active   High    Total     Total  PerMax   High
Total Entries           0        0       0        1       1       1
Trusted                 0        0       0        1       1       1
Blocked                0        0       0        1       1       1
```

ACL Operations		----- Lifetime -----		
	Recent	Total	PerMax	
ACL Requests	0	3	3	
Bad Messages	0	0	0	
Promotions	0	2	2	
Demotions	0	2	2	
Trust->Untrust	0	1	1	
Untrust->Deny	0	1	1	

show mgcp acls

The **show mgcp acls** command includes counters that track the number of endpoints demoted from trusted to untrusted and the number of endpoints demoted from untrusted to denied. For example:

```
ACMEPACKET# show mgcp acls
16:49:58-184
MGCP ACL Status      -- Period -- ----- 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 Operations      ----- Lifetime -----
                    Recent  Total  PerMax
ACL Requests         0       0     0
Bad Messages          0       0     0
Promotions            0       0     0
Demotions             0       0     0
Trust->Untrust       0       0     0
Untrust->Deny         0       0     0
```


Introduction

This section summarizes options for configuring the lawful intercept feature. It describes how the Net-Net SBC interoperates with mediation equipment from vendors who build lawful intercept equipment. If you are interested in the details of how this feature set works, refer to the *Net-Net LI Documentation Set*. There is one document available for each type of LI interoperability the Net-Net SBC supports.

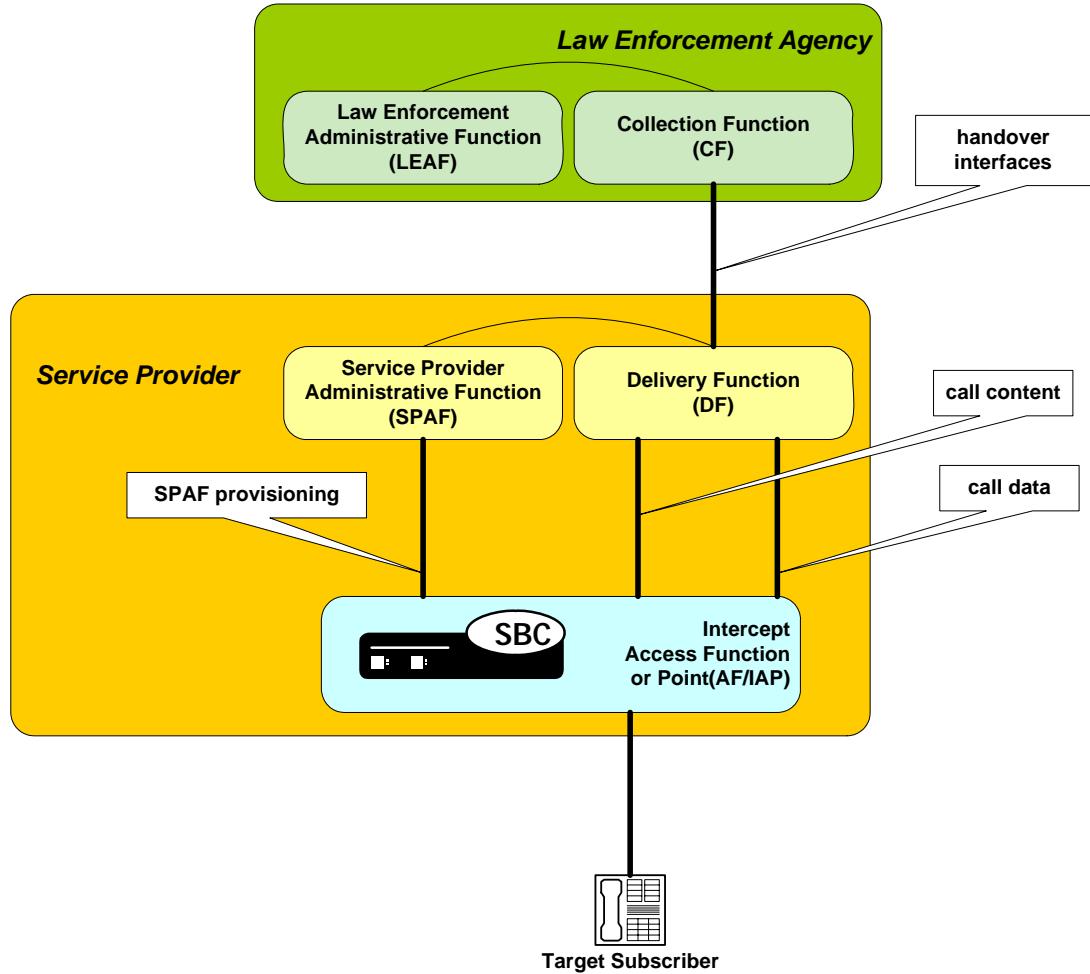
LI/CALEA consists of the interception of call content and/or the interception of call-identifying information. It requires that call information and media streams be sent to one or more law enforcement agencies in accordance with warrants for lawful interception.

You can configure your Net-Net SBC to support LI/CALEA functionality, enabling the Net-Net SBC to play a role in your Lawful Interception solution. Acting as an intercept access point (IAP), the Net-Net SBC can provide call data and can replicate media when signaling and media are anchored at the Net-Net SBC.

The Net-Net SBC supports LI/CALEA functionality that:

- Ensures unobtrusive intercept by hiding the network-based intercept of call information and content through topology hiding and media relay or NAT
- Intercepts and forwards call information and call content
- Interfaces with the mediation equipment [service provider administrative function (SPAF) and delivery function (DF)] for legal intercept

The following diagram provides one example of the Net-Net SBC deployed as part of a service provider's lawful intercept solution.



Recommendations

Calls may be lawfully intercepted by different devices in the service provider's network based on specific call flows, involvement of the device in the invoked service and where devices sit in the flow. Acme Packet recommends that you contact our Professional Services department to plan your use of the lawful intercept feature on your Net-Net SBC. Acme Packet Professional services can assist you with network/call flow analysis to determine which types of calls will involve the Net-Net SBC as an intercept access point and to recommend proper configuration.

Interoperability with SS8

The Net-Net SBC is configured to communicate with a trusted SS8 Xcipio SSDF for provisioning of target numbers by the SSDF and for delivery of call data (or call data and call content) by the Net-Net SBC to the SSDF.

The provisioning interface (INI-1) used between the Net-Net SBC and the SSDF is specified by SS8 in their Generic Interface (SS8 GI). Through this interface the Net-

Net SBC is provisioned with target numbers and is informed if the lawful interception is for call data only, or call data and call content replication.

For the purpose of call identification, call data events that are delivered by the Net-Net SBC over the CDC (INI-2) to the SSDF. The interface used for CD between the Net-Net SBC and the SSDF is the SS8 GI specification, which identifies mapping of SIP messages to call data events to be delivered over the CD interface.

The Net-Net SBC can intercept the content of calls without the subscriber being able to detect any change and without introducing any additional latency. The Net-Net SBC can duplicate the content and deliver replicated media over the CCC (INI-3) to the SSDF. The interface used for CC between the Net-Net SBC and the SSDF is compliant with PacketCable specification PKT-SP-ESP-I03-040113 (encapsulation in UDP).

The DF routes the call data and call content to the law enforcement agency over standards-based handover interfaces to the Collection Function (CF).

Interoperability with Verint

The Net-Net SBC is configured to communicate with a trusted Verint Systems STAR-GATE platform for provisioning of target numbers by STAR-GATE and for delivery of call data (or call data and call content) by the Net-Net SBC to STAR-GATE.

The provisioning interface (INI-1) used between the Net-Net SBC and STAR-GATE is specified by Verint in their INI-1 specification. Through this interface the Net-Net SBC is provisioned with target numbers and is informed if the lawful interception is for call data only, or call data and call content replication.

For the purpose of call identification, call data events that are delivered by the Net-Net SBC over the CDC (INI-2) to STAR-GATE. The interface used for CDC between the Net-Net SBC and STAR-GATE is PacketCable PKT-SP-EM-I08-040113 (RADIUS/ UDP, with Verint extensions), which identifies mapping of SIP messages to call data events to be delivered over the CD interface.

The Net-Net SBC can intercept the content of calls without the subscriber being able to detect any change and without introducing any additional latency. The Net-Net SBC can duplicate the content and deliver replicated media over the CCC (INI-3) to STAR-GATE. The interface used for CC between the Net-Net SBC and STAR-GATE is compliant with PacketCable specification PKT-SP-ESP-I03-040113 (encapsulation in UDP).

The DF routes the call data and call content to the law enforcement agency over standard-based handover interfaces to the Collection Function (CF).

Interoperability Using a Dynamic Trigger by CMS

The Net-Net SBC is configured to communicate with a trusted CMS (call management server) to enable dynamic lawful interception through the use of an intercept trigger for a specific SIP call and the delivery of call content by the Net-Net SBC to the DF. In this model, the CMS provides call identifying information to the DF.

A dynamic trigger is used by the CMS to signal the Net-Net SBC to duplicate call content and to deliver the replicated content to a specified DF.

The Net-Net SBC can intercept the content of calls without the subscriber being able to detect any change and without introducing any additional latency. The Net-Net

SBC can duplicate the content and deliver replicated media over the CCC (INI-3) to the specified DF. The interface used for CC between the Net-Net SBC and the specified DF must be compliant with PacketCable specification PKT-SP-ESP-I03-040113 (encapsulation in UDP).

The DF routes the call data (provided by the CMS) and call content (provided by the Net-Net SBC) to the law enforcement agency over standard-based handover interfaces to the Collection Function (CF).

Interoperability Using ALIP

The Net-Net SBC supports a lawful interface called Acme Packet Lawful Intercept Provisioning (ALIP).

The Net-Net SBC is configured to communicate with a trusted mediation device to enable dynamic lawful interception through the use of an intercept trigger for a specific SIP call and the delivery of call content by the Net-Net SBC to the DF. In this model, device provides call identifying information to the DF.

A dynamic trigger is used by the mediation equipment to signal the Net-Net SBC to duplicate call content and to deliver the replicated content to a specified DF.

The Net-Net SBC can intercept the content of calls without the subscriber being able to detect any change and without introducing any additional latency. The Net-Net SBC can duplicate the content and deliver replicated media over the CCC (INI-3) to the specified DF. The interface used for CC between the Net-Net SBC and the specified DF must be compliant with PacketCable specification PKT-SP-ESP-I03-040113 (encapsulation in UDP).

The DF routes the call data (provided by another device) and call content (provided by the Net-Net SBC) to the law enforcement agency over standard-based handover interfaces to the Collection Function (CF).

Interoperability Using X1, X2, X3

This document describes how the Net-Net SBC supports X1, X2, and X3 interfaces for lawful interception of SIP calls. In this deployment, the Net-Net SBC acts as an interception point that receives provisioning information from an administrative function and, based on that information, provides call data and content. As with the other LI interoperability solutions that the Net-Net SBC supports, the X1, X2, and X3 interfaces ensure unobtrusive call intercept by hiding network-based intercept of call data and content. The Net-Net SBC supports intercept of call data only, or of call data and call content.

Introduction

The Common Open Policy Service (COPS) [RFC 2748] is a protocol supported by the Net-Net SBC to perform and implement Call Admission Control (CAC) based on the policies hosted in an external policy server. While the Net-Net SBC already supports internal CAC policies, they are not as flexible as a Resource and Admission Control Function / Policy Decision Function (RACF/PDF), the generic resource and admission control functional architecture conceived by the ITU-T and the IETF.

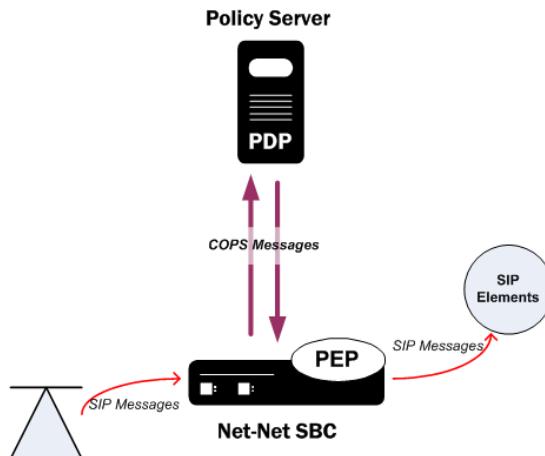
The Net-Net SBC COPS model includes a Policy server, functionally called the policy decision point (PDP), and the edge router, functionally called the policy enforcement point (PEP), the Net-Net SBC itself. The PDP and the PEP communicate with each other via the COPS protocol.

The Net-Net SBC also supports CLF services with its COPS implementation. Although the purpose of CLF is unlike the RACF/PDP functionality, COPS is the protocol the Net-Net SBC uses to talk to a CLF network device.

One of three licenses is required to use External Policy Server services: External Bandwidth Management-for RACF support; External CLF Mgmt-for CLF support; External Policy Services-for support of both RACF and CLF.

Call Admission Control

Admission control is performed according to the following typical scenario. When the Net-Net SBC receives a SIP INVITE, it sends a COPS request (REQ) message to the PDP. The REQ message includes the call ID, the SIP client's IP address, the Net-Net SBC's IP address and port number of the ingress interface for the call, and SDP based bandwidth requirements. The PDP responds with a COPS Decision (DEC) message with either the Install or Remove command. 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.



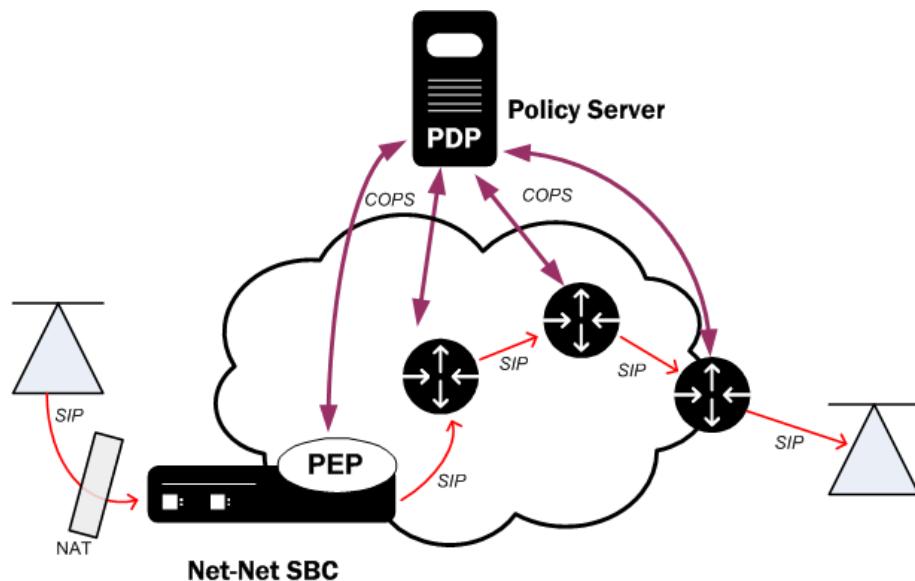
The Net-Net SBC can be configured so that both sides of a call, based on realm, are subject to COPS bandwidth enforcement. Each flow is treated as a unique call/event, because from a media and signaling perspective, they are distinct. As the Net-Net SBC functions as one side of a call, its IP address is inserted into the REQ message regardless of whether it is the calling or called party. This allows for the COPS 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 will delete the reservation on the PDP by sending a COPS delete request state (DRQ) message to the PDP. All ended calls must be deleted from the PDP in order to accurately track used and available bandwidth.

Implementation Features

As the Net-Net SBC proxies and forwards calls, caller media information is known before the INVITE reaches the callee. The PEP can request a specific amount of bandwidth for a call, and the PDP 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 to PEPs such as routers, along this path to the callee.

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 Session Director sends a request to the PDP requesting as much bandwidth as the codec with the highest bandwidth in the SDP message requires. If the call is

admitted, and when the called party returns finalized SDP information, the Session Director will modify the original reservation with the chosen codec's bandwidth requirements. This ensures the PDP has current and accurate information with which to make policy decisions.

COPS connection

The COPS session is established over a persistent TCP connection between the PDP and PEP. A COPS Client-Open (OPN) message is sent from the Net-Net SBC to the RACF, which responds with a COPS Client-Accept (CAT) message. A COPS Client-Close (CC) message is sent to either side to gracefully close the persistent connection. This COPS connection is expected to never close, unless an error occurs.

COPS Failures

Connection failures are discovered through a keep alive mechanism. Keep alive (KA) messages are periodically sent by the Net-Net SBC to the RACF regardless if any other COPS messages have been exchanged. When a KA message is not received, a connection failure is flagged. If the COPS connection fails, the Net-Net SBC will continually try to re-establish the connection to the PDP. Previously established calls will continue unaffected, but the Net-Net SBC will deny new calls from being established until the COPS connection is restored.

Failure Detection

A COPS connection failure is triggered by one of the three following events:

1. COPS KA timeout. The Net-Net SBC flags a COPS KA timeout when it does not receive a response for the KA it sent to the PDP. The PDP flags a COPS KA timeout when it does not receive the KA message within its requested timer time from the Net-Net SBC. At a minimum, when the COPS KA message times out, the TCP socket is closed.
2. Explicit COPS CC. The Net-Net SBC closes a COPS connection if it receives a COPS CC message from the PDP. The PDP closes a COPS connection if it receives a CC message from the Net-Net SBC. After the COPS layer connection is closed, then the TCP socket is closed too.
3. TCP socket termination. If either side receives a TCP FIN or RST, the TCP socket closes as expected. The COPS layer then detects that the socket has been closed before sending any further messages, and thus the COPS connection is closed.

Failure Recovery

The Net-Net SBC assumes that the PDP has a mechanism that re-uses the same logical IP Address, restarts itself in a timely manner, or fails over to another PDP. Therefore, no backup PDP IP address is configured on the Net-Net SBC.

The Net-Net SBC will try to re-open the COPS connection to recover from a connection failure. The PDP is never the device to initiate a connection. The Net-Net SBC increases its retry interval after successive reconnect failures. Once the retry interval has grown to every five minutes, the Net-Net SBC continues to retry to open the COPS connection at the five minute interval.

COPS PS Connection Down

You can configure whether or not you want the Net-Net SBC to reject or allow new calls to be established despite the failure of a policy server (PS) connection.

You enable this feature in the external policy server configuration using a new parameter. When you enable the feature, the Net-Net SBC allows new SIP calls to be established even though the connection to the PS has failed. In this case, the PS will not respond and will not be aware of the established sessions. When you disable

this feature, the Net-Net SBC behaves as it did in prior releases by responding to a connection failure with a 503 Service Unavailable.

Net-Net High Availability Support for COPS

The Net-Net SBC's high availability (HA) capabilities have been extended to support COPS. When one Net-Net SBC in an HA configuration goes down, the MAC addresses are reassigned to a healthy Net-Net SBC. IP addresses "follow" the MAC addresses to provide a seamless switchover between HA nodes.

After an HA failover, the COPS connection on the primary Net-Net SBC is either gracefully torn down, or times out depending on behavior of the PDP. The backup Net-Net SBC attempts to create a new COPS connection with the PDP. The OPN message uses the same PEPID and Client Type as in the previous pre-failover session.

COPS Debugging

A new argument has been added to the show command for viewing COPS and CAC statistics. From the user prompt, type **show <space> ext-band-mgr <return>**.

```
ACMEPACKET# show ext-band-mgr
10: 11: 38-194
EBM Status          -- Period -- ----- Lifetime -----
                    Active   High    Total     Total  PerMax   High
Client Trans        0       0       0       0       0       0
Server Trans        0       0       0       0       0       0
Sockets             1       1       1       1       1       1
Connections         0       0       0       0       0       0

----- Lifetime -----
Recent      Total  PerMax
Reserve      0       0       0
Modify       0       0       0
Commit       0       0       0
Remove       0       0       0
EBM Requests 0       0       0
EBM Installs 0       0       0
EBM Errors   0       0       0
EBM Rejects  0       0       0
EBM Expires  0       0       0
EBMD Errors  0       0       0
```

You can also refer to the **log.ebmd** log file located in the **/ramdrv/logs/** directory on the Net-Net SBC. This file must be retrieved via FTP or SFTP.

Configuring COPS

In the following configuration examples, we assume that your baseline configuration passes SIP traffic, with the Net-Net SBC in the role of an Access SBC. 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.

ACLI Instructions and Examples

To configure the realm configuration for COPS 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(real-m-config)#
4. Type **select** and the number of the pre-configured sip interface you want to configure.
ACMEPACKET(real-m-config)# **select 1**
ACMEPACKET(real-m-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 COPS 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-bw-manager**—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.

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)#[/ext-policy-server]

4. **name**—Enter the name for this external bandwidth manager instance. This parameter is used to identify the PDP used for that will be used in each Realm configuration.
5. **state**—Set state to **enabled** to run COPS. 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. This sets the COPS client type to **0x7926**. If another vendor's Policy Server is supported, it will be a different protocol value. The default value is **disabled**. The valid values are:
 - **disabled**—COPS is disabled
 - **admission-control**—Net-Net SBC acts as a PEP in a PDP/RACF deployment
 - **bandwidth-mgmt**—Net-Net SBC communicates with a CLF to obtain location string
7. **protocol**—Enter **COPS** to support COPS-based CAC. The **A-COPS** protocol implicitly sets the SD to use **0x4AC0** as the COPS client type. The default value is **C-SOAP**. The valid values are:
 - **COPS**—Standard COPS implementation. COPS client type is **0x7929** for CLF, and **0x7926** for PDP/RACF usage as defined in the operation-type parameter.
 - **A-COPS**—Vendor-specific protocol. COPS client type is **0x4AC0** for admission-control operation-type.
 - **SOAP**—Not used
 - **C-SOAP**—Not used
 - **DIAMETER**—Connects the Net-Net SBC to the policy-server
8. **address**—Enter the IP Address of the external COPS PDP.
9. **port**—Enter the port number the COPS connection connects to on the PDP. The default value is **80**. (The standard port for COPS is 3288.) 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 that the Net-Net SBC performs admission requests for.
11. **num-connections**—Enter the number of policy protocol TCP connections to establish to the PDP. The default value is **1**. The valid range is:
 - Minimum—0
 - Maximum—65535
12. **reserve-incomplete**—Set this parameter to **enabled** when communicating with a PDP via COPS. 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

13. **permit-conn-down**—Enter enabled for the Net-Net SBC to establish new SIP sessions despite PS connection failure. The default value is **disabled**. The valid values are:
 - enabled | disabled
14. Save your work using the ACLI **done** command.

```
ext-policy-server
  name          test-PDP
  state         enabled
  operation-type bandwidth-mgmt
  protocol      COPS
  address       192.168.40.50
  port          80
  realm         PDPrealm
  permit-conn-down disabled
  num-connections 1
  reserve-incomplete enabled
  last-modified-date 2006-01-31 15:38:07
ACMEPACKET(ext-policy-server)#

```

To configure the media profile configuration for COPS support in a CAC scenario:

Values for the following parameters can be found in the PacketCable™ Audio/Video Codecs Specification PKT-SP-CODEC-I06-050812 document.

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**
ACMEPACKET(media-profile)#
 4. Type **select** and the number of the pre-configured media profile you want to configure.
ACMEPACKET(media-profile)# **select 1**
ACMEPACKET(media-profile)#
 5. **peak-rate-limit**—Enter the r, P value:
 - **r**—bucket rate
 - **p**—peak rate
 6. **max-burst-size**—Enter the b, m, M value:
 - **b**—Token bucket size
 - **m**—Minimum policed unit
 - **M**—Maximum datagram size
 7. Save your work using the ACLI **done** command.

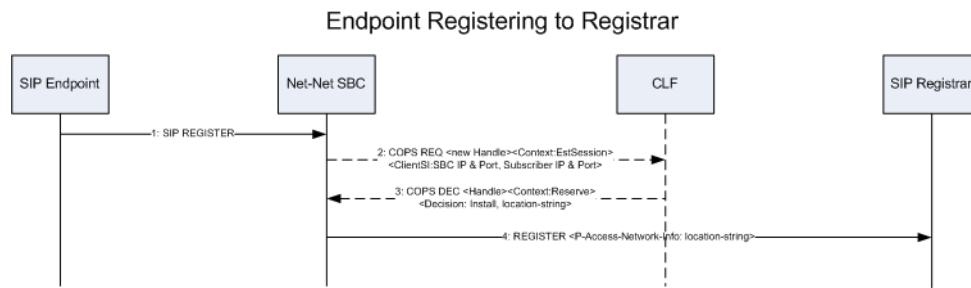
Connectivity Location Function

A Connectivity Location Function (CLF) maintains mappings between endpoints with dynamically assigned IP addresses and their physical location. The Net-Net SBC, acting as a P-CSCF, is the intermediary device between a registering endpoint and a CLF. The CLF thus validates and tags a registering endpoint, and the Net-Net SBC applies the CLF's actions. The Net-Net SBC and the CLF maintain a connection with each other using the COPS protocol.

CLF Behavior

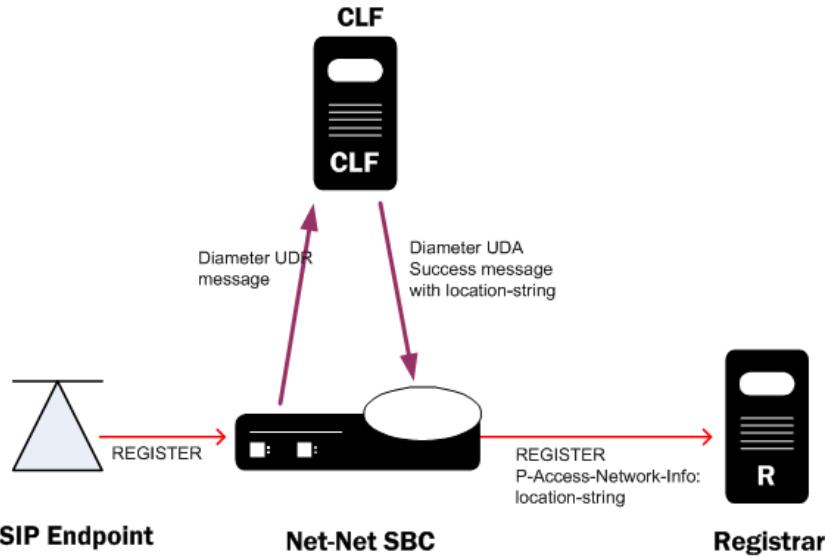
The Net-Net SBC and a CLF only interact when an endpoint registers or re-registers. The Net-Net SBC, acting as the P-CSCF, is the first SIP device that the REGISTER message reaches. Upon receiving the REGISTER message(1), the Net-Net SBC queries the CLF using the COPS protocol. The endpoint's (public) IP address and port, and the Net-Net SBC's IP information are sent to the CLF in a COPS REQ message(2).

The CLF responds to the Net-Net SBC with an Approve or Reject COPS DEC message(3). If the request is approved, then the CLF also sends a location-string value to be inserted in one of the SIP headers. The Net-Net SBC inserts a P-Access-Network-Info header containing the location-string into the incoming REGISTER message and forwards this message(4) to the SIP registrar/I/S-CSCF.



The Net-Net SBC will insert this P-Access-Network-Info header into all subsequent SIP messages from this endpoint as they are forwarded into the core network. The P-Access-Network-Info header is inserted into all SIP requests and responses except for ACK and CANCEL messages. For all boundaries where SIP messages pass from

trusted to untrusted SIP interfaces or session agents, the Net-Net SBC will strip out the P-Access-Network-Info header as expected.



If the CLF responds with a Reject DEC message, the Net-Net SBC rejects the registration, and sends a 503 - Service Unavailable message back to the registering endpoint. In this way, the CLF can be used for admission control.

The Net-Net SBC communicates with the CLF solely for retrieving location information from the CLF, and not for notifying the CLF about an endpoint's registration state or activity. When an endpoint's registration ends, either through a normal expiration, getting rejected by the registrar, or through specific de-registering or error conditions, the Net-Net SBC deletes the locally cached registration location string. The Net-Net SBC does not update the CLF about any registrations that have been deleted.

P-Access-Network-Info Header Handling

The P-Access-Network-Info header is created and populated according to the following rules:

1. If the CLF returns an Accept DEC message and a location string, the Net-Net SBC inserts the location string into a P-Access-Network-Info header in the outgoing REGISTER message.
2. If the CLF returns an Accept DEC message without a location string, the Net-Net SBC inserts the configured default string into a P-Access-Network-Info header in the outgoing REGISTER message.
3. If the CLF returns an Accept DEC message without a location string and no location string is configured on Net-Net SBC, the outgoing REGISTER message is forwarded out of the Net-Net SBC, but no P-Access-Network-Info header is created for the REGISTER message.

CLF Re-registration

The Net-Net SBC will send a new REQ message to the CLF to request a new location string if any of the following events occur:

1. The endpoint's contact address changes.
2. The SIP Register message's Call-ID header changes.
3. The endpoint's public IP Address or UDP port changes.

4. The endpoint connects to a different SIP interface, port, or realm on the Net-Net SBC than it did in the initial REGISTER message.
5. The registration expires in the Net-Net SBC's registration cache.

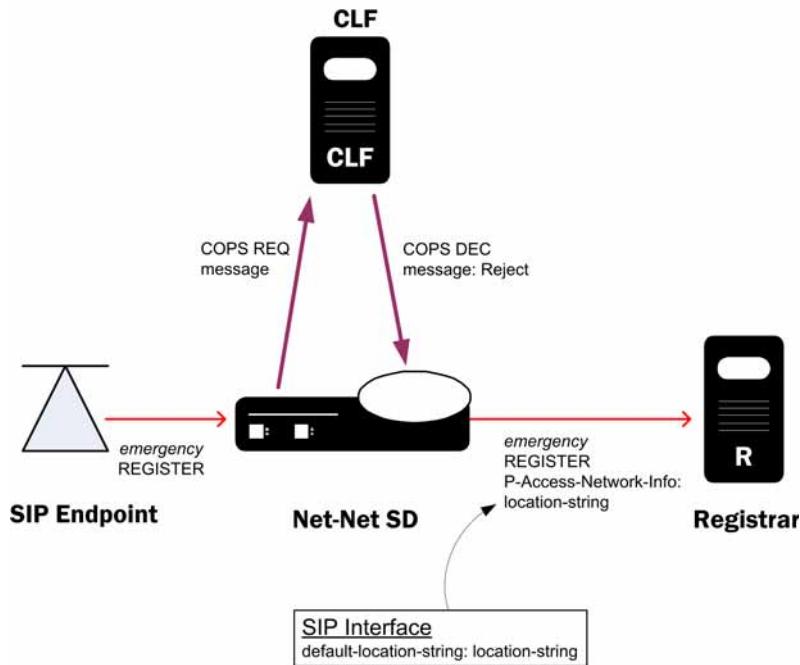
CLF Failures

If a COPS connection fails, the Net-Net SBC will continually try to re-establish the connection. Endpoints that are already registered will stay registered unless they timeout or if the registrar rejects their refreshes. When the COPS connection has not been established, and an endpoint registers on a SIP interface that is configured to use CLF, the Net-Net SBC forwards new REGISTER messages to the registrar using the default location string.

CLF Emergency Call Handling

The Net-Net SBC allows emergency calls into the network even if the endpoint that places the emergency call is not registered. In the expected fashion, the Net-Net SBC will query the CLF first for an incoming emergency call sourced from an unregistered endpoint. If the CLF response is successful, then the Net-Net SBC will insert the string returned from the CLF into a P-Access-Network-Info header, and insert this header into the emergency call's REGISTER message. If no location string is returned with a successful CLF response, the default location string is inserted into P-Access-Network-Info header.

If the CLF's response is to reject the emergency call, the Net-Net SBC will insert the configured default location string into the P-Access-Network-Info header and forward the emergency call's REGISTER message toward the registrar. For emergency calls where the endpoint has already successfully registered, the call will be routed into the network using the expected methods for emergency call routing.



If the COPS connection to the CLF is down, emergency calls from un-registered endpoints are still allowed into the network using the default string inserted into the emergency messages.

HA Functionality

The location strings generated by the CLF are replicated on the standby SD in an HA pair. This is required so that a Net-Net SBC in an HA pair can instantly continue processing calls using the previously learned CLF information.

CLF Debugging

A new argument has been added to the show command for viewing CLF statistics. From the user prompt, type `show <space> ext-clf-svr <return>`.

```
ACMEPACKET# show ext-clf-svr
14:17:14-114
EBM Status
----- Period ----- Lifetime -----
Active High Total Total PerMax High
Client Trans 0 0 0 0 0 0
Server Trans 0 0 0 0 0 0
Sockets 0 0 0 0 0 0
Connections 0 0 0 0 0 0

----- Lifetime -----
Recent Total PerMax
CLF Requests 0 0 0
CLF Admits 0 0 0
CLF Errors 0 0 0
CLF Rejects 0 0 0
CLF Expires 0 0 0
CLFD Errors 0 0 0
```

You can also refer to the `log.ebmd` log file located in the `/ramdrv/logs/` directory on the Net-Net SBC. This file must be retrieved via FTP or SFTP.

Configuring CLF

In the following configuration examples, we assume that your baseline configuration passes SIP traffic, with the Net-Net SBC in the role of an Access SBC. In this example, you will configure additions to the realm configuration and the new external policy server configuration.

ACL Instructions and Examples

To configure the SIP interface configuration for CLF 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-related configurations.

```
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(configure)# sip-interface
ACMEPACKET(sip-interface)#

```
4. Type `select` and the number of the pre-configured sip interface you want to configure for CLF. This should be the ingress SIP interface for

```
ACMEPACKET(sip-interface)# select 1
```

```
ACMEPACKET(sip-interface) #
```

5. **ext-policy-svr**—Set this parameter to the same name as the External Policy Server configured that you configured for the CLF server.
6. **default-location-string**—Set this parameter to the default location string you want inserted into a P-Access-Network-Info header for when the CLF server does not return a unique location string.
7. Save your work using the ACLI **done** command.

To configure the external policy server for use with a CLF:

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**—Set this parameter to an applicable name for this CLF instance of the external policy server. The value of this parameter will be entered in the SIP interface configuration element to reference this CLF.
5. **state**—Set this parameter to **enabled** to enable this CLF. The default value is **enabled**. The valid values are:
 - enabled | disabled
6. **operation-type**—Set this parameter to **admission-control** for the Net-Net SBC to communicate with a CLF. The default value is **disabled**. The valid values are:
 - **disabled**—Disable this parameter.
 - **admission-control**—Net-Net SBC acts as a PEP in a PDP/RACF deployment
 - **bandwidth-mgmt**—Net-Net SBC communicates with CLF to obtain location string
7. **protocol**—Set this parameter to **COPS** to connect with a CLF via the COPS protocol. The default value is **C-SOAP**. The valid values are:
 - **COPS**—Standard COPS implementation. COPS client type is 0x7929 for CLF, and 0x7926 for PDP/RACF usage as defined in the operation-type parameter.
 - **A-COPS**—Vendor specific protocol. COPS client type is 0x4AC0 for admission-control operation-type.
 - **SOAP**—Not used
 - **C-SOAP**—Not used
 - **DIAMETER**—Connects the Net-Net SBC to the policy-server
8. **address**—Set this parameter to the IP address of the CLF.
9. **port**—Set this parameter to the port which the CLF uses for COPS transactions. The standard port for COPS is **3288**. The default value is **80**. The valid range is:
 - Minimum—0

- Maximum—65535
10. **realm**—Set this parameter to the realm in which the CLF exists.
 11. **num-connections**—Set this parameter to the number of connections the Net-Net SBC will create with the CLF. The default value is 1. The valid range is:
 - Minimum—0
 - Maximum—65535
 12. **reserve-incomplete**—Set this parameter to **enabled** if you want the Net-Net SBC to send a COPS REQ message to the CLF that does not include the endpoint's true port number. A value of 0 will be used for the port number. The default value is **enabled**. The valid values are:
 - enabled | disabled
 13. Save your work using the ACLI **done** command.

```
ACMEPACKET(ext-pol i cy-server)# show
ext-pol i cy-server
  name          testCLF
  state         enabl ed
  operati on-type admi ssi on-control
  protocol      COPS
  address       192. 168. 50. 40
  port          80
  real m        cl f-real m
  num-connecti ons 1
  reserve-i ncompl ete enabl ed
```

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) and Connectivity Location Function (CLF) applications. The existing licenses for COPS based CLF and RACF support Diameter and COPS.

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 and CLF are available from the front media interfaces 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.

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.

A Diameter connection failure is determined by one of the three events:

1. Diameter Device-Watchdog timeout—The Net-Net SBC detects a timeout when it does not receive a DWR from the Diameter server within the guard timer period. When this happens, the Net-Net SBC tears down the TCP connection and attempts to reconnect to the failed Diameter server.
2. TCP socket termination—if either side of the Diameter connection receives a FIN or RST, the TCP socket closes per standard behavior. The Net-Net SBC periodically tries to reconnect to the Diameter server.

Application IDs and Modes

Diameter messages include an application ID to indicate the application and standards' body interface. The following table lists the different Application-IDs for the corresponding standards' and applications. Application IDs must be provisioned manually.

Standards Reference Point				
	RACF		CLF	
AVP	Gq 3GPP R6 29.209	Rx 3GPP R7 29.214	Rq ETSI 283 026	e2 ETSI 283 035
Application-ID	16777222	16777229	16777222	16777231

Note: For CLF application, the Application-ID should be set to the value indicated in the table above.

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 tell the Net-Net SBC the format for DIAMETER messages it sends and receives.

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	<p>The interface runs in Rx mode when you either:</p> <ul style="list-style-type: none"> 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	<p>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.</p>
e2	<p>The interface runs in e2 mode, the base CLF interface, when you set the application mode to e2. Even if you leave the application mode set to none, the interface will run in e2 mode when the external policy server is configured as a CLF interface.</p>
none	<p>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.</p>

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.

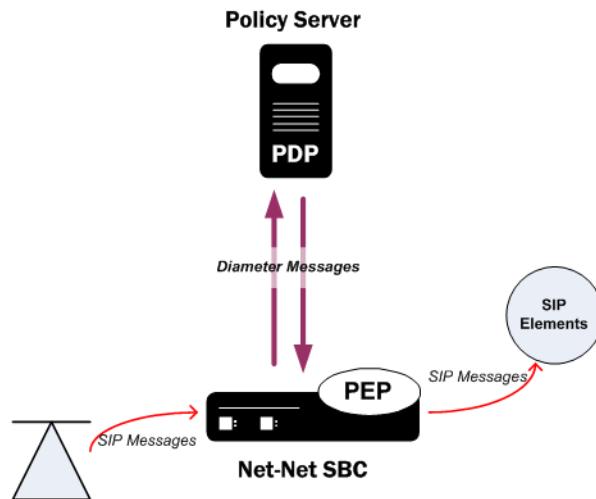
Implementation Features

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

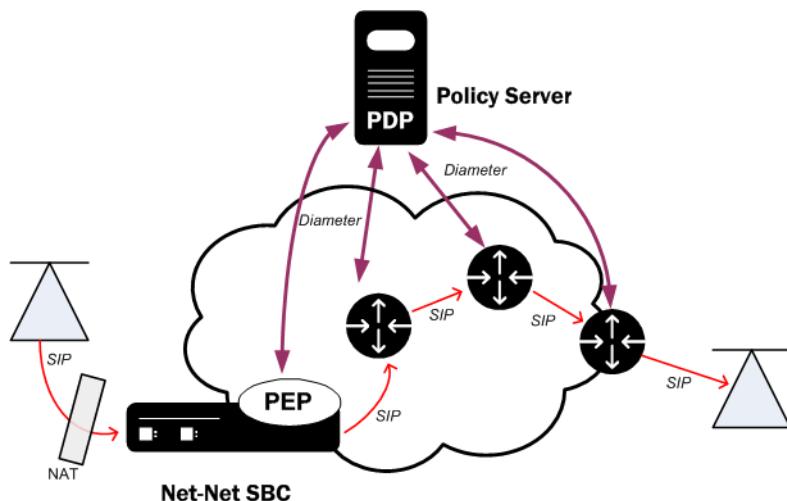
SBC to allow all calls when the RACF is unreachable if this is the desired behavior.



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 on the 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.

DIAMETER AAR Query Post SDP Exchange

For DIAMETER, the Net-Net SBC supports sending the Authentication-Authorize-Request (AAR) query upon SDP answer instead of the SDP offer. This change can be useful in WiMax environments where mobile phones go idle and become semi-detached from their base stations and from the WiMax access controller (WAC). In such a case, the WAC receives an AAR from the idle user but, because it cannot determine that user's base station, rejects the request.

You enable this behavior by setting the **reserve-incomplete** parameter to **origin-realm-only**.

Net-Net High Availability Support for CAC

The Net-Net SBC's high availability (HA) capabilities support CAC. When one Net-Net SBC in an HA configuration goes out of service, the MAC addresses are reassigned to a healthy Net-Net SBC. IP addresses follow the MAC addresses to provide a seamless switchover between HA nodes.

After an HA failover, the Diameter connection on the primary Net-Net SBC is either gracefully torn down, or times out depending on behavior of the PDP. The backup Net-Net SBC attempts to create a new Diameter connection with the PDP.

ACLI Instructions and Examples

In the following configuration examples, we assume that your baseline configuration passes SIP traffic, with the Net-Net SBC in the role of an Access SBC. 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 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# **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(real-m-config)#
4. Type **select** and the number of the pre-configured sip interface you want to configure.
ACMEPACKET(real-m-config)# **select 1**
ACMEPACKET(real-m-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-bw-manager**—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.

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# **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 CAC. 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
 - **admission-control**—Net-Net SBC acts as a PEP in a PDP/RACF deployment
 - **bandwidth-mgmt**—Net-Net SBC communicates with a CLF to obtain location string
7. **protocol**—Enter **Diameter** to support Diameter-based CAC. The default value is **C-SOAP**. The valid values are:
 - **COPS**—Standard COPS implementation. COPS client type is 0x7929 for CLF, and 0x7926 for PDP/RACF usage as defined in the operation-type parameter.
 - **A-COPS**—Vendor specific protocol. COPS client type is 0x4AC0 for admission-control operation-type.
 - **SOAP**—Not used
 - **C-SOAP**—Not used
 - **DIAMETER**—Connects the Net-Net SBC to the policy-server
8. **address**—Enter the IP Address of the external PDP.
9. **port**—Enter the port number the diameter connection connects to on the PDP. The standard port for COPS is **3288**. The default value is **80**. 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. **product-name**—Enter text string that describes the vendor-assigned name for the RACF. This parameter is required.
13. **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 \(1018\)](#). The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
14. **application-id**—Enter the type of interface you want to use. Your choices are: **Rq**, **Rx**, **Gq**, **e2**, and **none**. For details about each type, refer to this chapter's [Application IDs and Modes \(1018\)](#).
15. **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
16. **num-connections**—Enter the number of policy protocol TCP connections to establish to the PDP. For Diameter, this should be the value **1**. The default value is **1**. The valid range is:
- Minimum—**0**
 - Maximum—**65535**
17. **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**
18. **watchdog-ka-timer**—Enter the interval in seconds used to determine a DWA response timeout.
19. **reserve-incomplete**—Set this parameter to **enabled** when communicating with a PDP via Diameter. The parameter allows the Session Director 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** (default)—This mode supports the usual behavior when the AAR is sent upon SDP offer as well as SDP answer. This mode ensures backwards compatibility.
 - **orig-realm-only**—This mode allows calls originating from a realm with a policy server associated with it to send the AAR upon SDP offer. However, calls terminating at a realm with a policy server associated with it send the AAR post SDP exchange.
 - **disabled**—This mode allows no bandwidth reservation for incomplete flows.
20. Save your work using the ACLI **done** command.

```
ext-pol icy-server
      name          server5
      state         enabled
      operation-type bandwidh-mgmt
      protocol      DIAMETER
      address       1.0.11.12
      port          80
      realm         realm1
      permit-conn-down disabled
      product-name  SRACF-Policy-Srvr
      application-id 0
      framed-ip-addr-encoding octet-string
      num-connections 500
      reserve-incomplete enabled
```

Configuring Media Profiles for Diameter Support: CAC Scenario

Values for the following parameters can be found in the PacketCable™ Audio/Video Codecs Specification PKT-SP-CODEC-I06-050812 document.

To configure the media profile configuration for Diameter support in a CAC scenario:

1. In Superuser mode, type **configure terminal** and press <Enter>
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the session router path.
ACMEPACKET(config)# 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. Type **select** and the number of the pre-configured media profile you want to configure.
ACMEPACKET(media-profile)# select 1
ACMEPACKET(media-profile)#
5. **req-bandwidth**—Enter the required bandwidth in Kbps for the selected media profile. This is the bandwidth that the SBC will request from the policy server. The default value is zero (0). The valid values are:
 - Minimum—0
 - Maximum— $2^{32}-1$
6. Save your work using the ACLI **done** command.

Subscriber Information AVP

Certain policy servers rely on having the user's URI information available as means to identify the endpoint/subscriber. In addition to conveying the L3 IP address of a user, the Net-Net SBC supports RFC 4009: The Subscription-Id AVP. It identifies the end user's subscription and is used in 3GPP Rx reference point. This feature can be enabled regardless of the selected application mode of the external policy server.

Subscription ID AVP

The Subscription-Id AVP (AVP Code 443) includes a Subscription-Id-Data AVP that holds the identifier and a Subscription-Id-Type AVP that defines the identifier type. The external policy server configuration element is configured with an option to enable sending the Subscription-Id AVP to the policy server in an AA-Request message.

Subscription-Id-Type

The Net-Net SBC supports two Subscription-Id-Types:

Value	Name	Description
0	END_USER_E164	Identifier is in international E.164 format (e.g., MSISDN), according to the ITU-T E.164 numbering plan.
2	END_USER_SIP_URI	Identifier is in the form of a SIP URI.

The Subscription type that the Net-Net SBC sends in an AA-Request message depends on the received message's request line's Request URI. If the request line's URI is in SIP or SIPS format, then the AA-Request message indicates Subscription-Id-Type of 2.

If the request line's Request URI is in E.164 format, then the AA-Request message indicates Subscription-Id-Type of 0.

Subscription-Id-Data

There are two cases that determine what information is used for the Subscription-Id-Data value. In the following 2 cases, the external policy server is associated with the access realm.

1. When the Net-Net SBC receives a message originating in the access realm, the Subscription-Id-Data value is the URI in the **From:** header of the incoming SIP message.
2. When the Net-Net SBC receives a message originating in the core realm, the Subscription-Id-Data value is the URI in the **Request-URI:** header (identifying the [To:]called subscriber) of the incoming SIP message.

If the Net-Net SBC receives a message with its request URI in SIP/SIPS format, it sends the SIP Request URI as the Subscription-Id-Data. If the Net-Net SBC receives a message with its request URI in E.164 format, it sends the TEL Request URI as the Subscription-Id-Data.

ACLI Instructions and Example**To configure the Net-Net SBC to send Subscription ID AVP in AA-Request Messages:**

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```

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

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

3. Type **ext-policy-server** and press <Enter>.

```
ACMEPACKET(session-router)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```

If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the external policy server that you want to edit.

4. **options**—Set the options parameter by typing **options**, a <Space>, the option-name **include-sub-info** with a “plus” sign in front of it, and then press <Enter>.

```
ACMEPACKET(ext-policy-server)# options +include-sub-info
```

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 policy server 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.

CAC Debugging

A new argument has been added to the **show** command for viewing CAC statistics. From the user prompt, type **show <space> ext-band-mgr <return>**.

```
ACMEPACKET# show ext-band-mgr
10:11:38-194
EBM Status          -- Period -- ----- Lifetime -----
                    Active   High    Total      Total  PerMax   High
Client Trans        0       0       0         0       0       0
Server Trans        0       0       0         0       0       0
```

Sockets	1	1	1	1	1	1
Connections	0	0	0	0	0	0
----- Lifetime -----						
	Recent	Total	PerMax			
Reserve	0	0	0			
Modify	0	0	0			
Commit	0	0	0			
Remove	0	0	0			
EBC Requests	0	0	0			
EBC Installs	0	0	0			
EBC Errors	0	0	0			
EBC Rejects	0	0	0			
EBC Expires	0	0	0			
EBMD Errors	0	0	0			

You can also refer to the `log.ebmd` log file located in the `/ramdrv/logs/` directory on the Net-Net SBC. This file must be retrieved via FTP or SFTP.

Diameter: CLF

The Diameter base protocol (RFC 3588) is supported by the Net-Net SBC and is used for Bandwidth-Based Call Admission Control (CAC) and Connectivity Location Function (CLF) applications. The existing licenses for COPS based CLF and RACF support Diameter and COPS.

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 and CLF are available from the front media interfaces 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.

Rx Interface Details

You now configure the Diameter interface on your Net-Net SBC to run in Rx mode. When you set the appropriate **application-id** and **application-mode** parameters, the Net-Net SBC runs in Rx mode because this value identifies this as the Rx interface. For all Authentication-Authorization-Request messages the Net-Net SBC sends over its Diameter interface, and:

- The Reservation-Priority AVP is included for priority calls. This AVP will be the main AVP within the AAR message.
- The Codec-Data AVP is then included for non-priority calls. This AVP is one of several that together comprise a Group AVP structure.

Non-Priority Call Handling

When a SIP signaling event triggers external bandwidth management use, the Net-Net SBC removes all SDP information from the signaling message that was the trigger. The Net-Net SBC repackages this bandwidth information so that it can form a Bandwidth Request and decide on an external bandwidth manager to which it should be sent. If the appropriate external bandwidth manager is configured for Rx

interface use, then Net-Net SBC then reformats the SDP information to construct a Codec-Data AVP.

If the external bandwidth manager that receives the request ignores the SDP information, then it does not include the Codec-Data AVP in the AAR.

For calls that do not require special treatment, the Codec-Data AVP is required to have the:

- AVP code 524
- 3GPP vendor identification number (10415)
- "V" (vendor) bit set in the AVP
- "M" (mandatory) bit set when sending this AVP
- Type octet string

In addition, the Codec-Data AVP must be encoded as described in the following table.

AVP section/line	Requirement
Line 1	Must specify the direction of the flow by including the ASCII "uplink" or "downlink": <ul style="list-style-type: none"> • uplink—Identifies the SDP as having come from the UE and sent to the network • downlink—Identifies the SDP as having come from the network and sent to the UE
Line 2	Must specify whether the offer or answer codec is at issue by including the ASCII "offer" (from an SDP offer according to RFC 3264) or "answer" (from an answer according to RFC 3264)
Remainder of the AVP	Must include lines found in the signaling SDP, formatted in ASCII and separated by new-line characters; the first line of this section must be the "m" line, followed by any "a" or "b" lines related to that "m" line

Priority Call Handling

The Net-Net SBC determines that a call is priority call when it either matches a defined network management control (NMC) priority rule. No other scenario triggers the priority call handing treatment described in this section.

When a SIP signaling event triggers external bandwidth management use for a priority call, the Net-Net SBC forms the Band Request reflecting the call's priority status and determines which external bandwidth manager to use. If the appropriate external bandwidth manager is configured for Rx interface use, then Net-Net SBC then reformats the SDP information to constructs a Reservation-Priority AVP and includes it in the AAR message.

For priority calls, the Reservation-Priority AVP is included in the AAR and is required to:

- Use the ETSI Vendor identification number (13019)
- Have the "V" (vendor) bit set in the AVP
- Not to have the "M" (mandatory) bit set when sending this AVP
- Required to be of the type enumeration with the following possible values: 0—default or 1 though 7—Priorities one through seven, where the level of priority increases with numeric value

For the Net-Net SBC, the Reservation-Priority AVP will be set to PRIORITY-SEVEN (7) if it is present.

Gq Interface

The Net-Net SBC supports the Gq interface (16777222) for Diameter protocol. As an enhancement to supporting this interface, the Net-Net SBC offers proxy bit enhancements to determine where Diameter messages should be handled.

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.

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.

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.

A Diameter connection failure is determined by one of the three events:

1. Diameter Device-Watchdog timeout—The Net-Net SBC detects a timeout when it does not receive a DWR from the Diameter server within the guard timer period. When this happens, the Net-Net SBC tears down the TCP connection and attempts to reconnect to the failed Diameter server.
2. TCP socket termination—if either side of the Diameter connection receives a FIN or RST, the TCP socket closes per standard behavior. The Net-Net SBC periodically tries to reconnect to the Diameter server.

Please see the [Diameter Heartbeat \(1017\)](#) section for information about configuring Diameter heartbeat.

Diameter: Connectivity Location Function

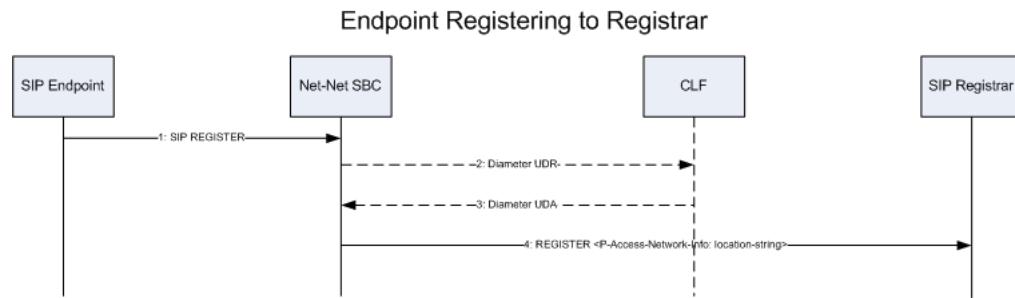
A Connectivity Location Function (CLF) maintains mappings between endpoints with dynamically assigned IP addresses and their physical location. The Net-Net SBC, acting as a P-CSCF, is the intermediary device between a registering endpoint and a CLF. The CLF thus validates and tags a registering endpoint, and the Net-Net

SBC applies the CLF's actions. The Net-Net SBC supports both COPS and Diameter protocols to maintain a connection with the CLF.

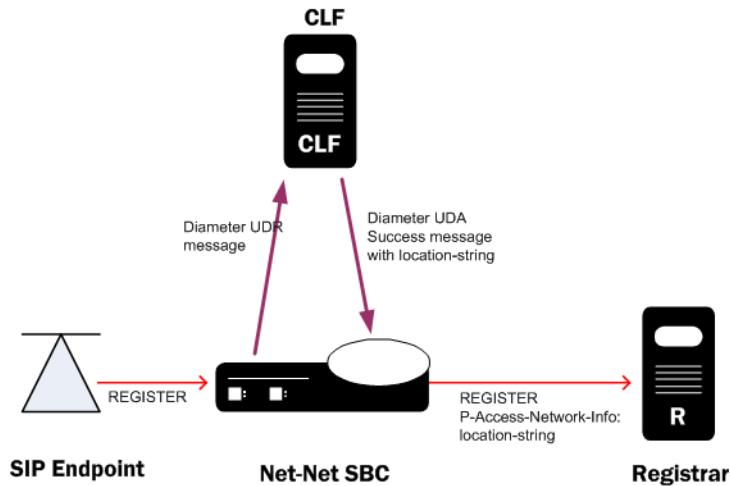
CLF Behavior

The Net-Net SBC and a CLF only interact with each other when an endpoint registers or re-registers. The Net-Net SBC, acting as the P-CSCF, is the first SIP device that the REGISTER message reaches. Upon receiving the REGISTER message(1), the Net-Net SBC queries the CLF using the Diameter protocol. The endpoint's (public) IP address and port, and the Net-Net SBC's IP information are sent to the CLF in a Diameter User-Data-Request (UDR) message(2).

The CLF responds to the Net-Net SBC with a Diameter User-Data-Answer (UDA) message(3). If the request is approved, then the CLF also sends a location-string value to be inserted in one of the SIP headers. The Net-Net SBC inserts a P-Access-Network-Info header containing the location-string into the incoming REGISTER message and forwards this message(4) to the SIP registrar/I/S-CSCF.



The Net-Net SBC inserts this P-Access-Network-Info header into all subsequent SIP messages from this endpoint as they are forwarded into the core network. The P-Access-Network-Info header is inserted into all SIP requests and responses except for ACK and CANCEL messages. For all boundaries where SIP messages pass from trusted to untrusted SIP interfaces or session agents, the Net-Net SBC will strip out the P-Access-Network-Info header as expected.



If the CLF responds with a Reject UDA message, the Net-Net SBC rejects the registration, and sends a 503 - Service Unavailable message back to the registering endpoint. In this way, the CLF can be used for admission control.

The Net-Net SBC communicates with the CLF solely for retrieving location information from the CLF, and not for notifying the CLF about an endpoint's registration state or activity. When an endpoint's registration ends, either through a normal expiration, getting rejected by the registrar, or through specific de-registering or error conditions, the Net-Net SBC deletes the locally cached registration location string. The Net-Net SBC does not inform the CLF about any registrations that have been deleted.

P-Access-Network-Info Header Handling

The P-Access-Network-Info header is created and populated according to the following rules:

1. If the CLF returns an Accept UDA message with a location string, the Net-Net SBC inserts the location string into a P-Access-Network-Info header in the outgoing REGISTER message.
2. If the CLF returns an Accept UDA message without a location string, the Net-Net SBC inserts the configured default string into a P-Access-Network-Info header in the outgoing REGISTER message.
3. If the CLF returns an Accept UDA message without a location string and no location string is configured on Net-Net SBC, the outgoing REGISTER message is forwarded out of the Net-Net SBC, but no P-Access-Network-Info header is created for the REGISTER message.

CLF Re-registration

The Net-Net SBC will send a new UDR message to the CLF to request a new location string if any of the following events occur:

1. The endpoint's contact address changes.
2. The SIP Register message's Call-ID header changes.
3. The endpoint's public IP Address or UDP port changes.
4. The endpoint connects to a different SIP interface, port, or realm on the Net-Net SBC than it did in the initial REGISTER message.
5. The registration expires in the Net-Net SBC's registration cache.

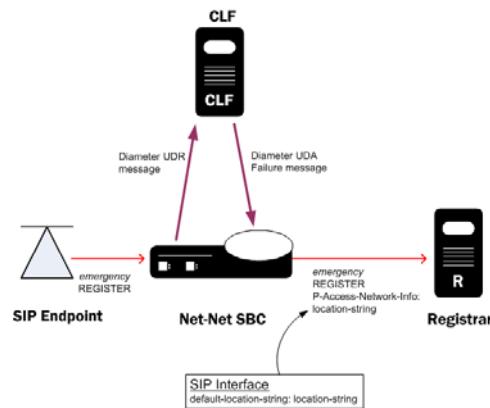
CLF Failures

If a Diameter connection fails, the Net-Net SBC will continually try to re-establish the connection. Endpoints that are already registered will stay registered unless they timeout or if the registrar rejects their refreshes. When the Diameter connection has not been established, and an endpoint registers on a SIP interface that is configured to use CLF, the Net-Net SBC forwards new REGISTER messages to the registrar using the default location string.

CLF Emergency Call Handling

The Net-Net SBC allows emergency calls into the network even if the endpoint that places the emergency call is not registered. In the expected fashion, the Net-Net SBC will query the CLF first for an incoming emergency call sourced from an unregistered endpoint. If the CLF response is successful, then the Net-Net SBC will insert the string returned from the CLF into a P-Access-Network-Info header, and insert this header into the emergency call's REGISTER message. If no location string is returned with a successful CLF response, the default location string is inserted into P-Access-Network-Info header.

If the CLF's response is to reject the emergency call, the Net-Net SBC will insert the configured default location string into the P-Access-Network-Info header and forward the emergency call's REGISTER message toward the registrar. For emergency calls where the endpoint has already successfully registered, the call will be routed into the network using the expected methods for emergency call routing.



If the Diameter connection to the CLF is down, emergency calls from un-registered endpoints are still allowed into the network using the default string inserted into the emergency messages.

HA Functionality

The location strings generated by the CLF are replicated on the standby SBC in an HA pair. This is required so that a Net-Net SBC in an HA pair can instantly continue processing calls using the previously learned CLF information.

ACLI Instructions and Examples

In the following configuration examples, we assume that your baseline configuration passes SIP traffic, with the Net-Net SBC in the role of an Access SBC. In this example, you will configure additions to the realm configuration and the new external policy server configuration.

SIP Interface Configuration for CLF Support

To configure the SIP interface configuration for CLF support:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# **config 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>. 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 for CLF. This should be the ingress SIP interface for
ACMEPACKET(sip-interface)# **select 1**
ACMEPACKET(sip-interface)#
 5. **ext-policy-svr**—Set this parameter to the same name as the External Policy Server configured that you configured for the CLF server.

6. **default-location-string**—Set this parameter to the default location string you want inserted into a P-Access-Network-Info header for when the CLF server does not return a unique location string.
7. Save your work using the ACLI **done** command.

External Policy Server for Use with a CLF

To configure the external policy server for use with a CLF:

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **media-manager** and press <Enter> to access the media-related configurations.
ACMEPACKET(config)# 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**—Set this parameter to an applicable name for this CLF instance of the external policy server. The value of this parameter will be entered in the SIP interface configuration element to reference this CLF.
5. **state**—Set this parameter to **enabled** to enable this CLF. The default value is **enabled**. The valid values are:
 - enabled | disabled
6. **operation-type**—Set this parameter to **admission-control** for the Net-Net SBC to communicate with a CLF. The default value is **disabled**. The valid values are:
 - **disabled**—This parameter is disabled
 - **admission-control**—Net-Net SBC acts as a PEP in a PDP/RACF deployment
 - **bandwidth-mgmt**—Net-Net SBC communicates with a CLF to obtain location string
7. **protocol**—Set this parameter to **DIAMETER** to connect with a CLF via the DIAMETER protocol. The default value is **C-SOAP**. The valid values are:
 - **COPS**—Standard COPS implementation. COPS client-type is 0x7929 for CLF, and 0x7926 for PDP/RACF usage as defined in the operation-type parameter
 - **A-COPS**—Vendor specific protocol. COPS client-type is 0x4AC0 for admission-control operation-type.
 - **SOAP**—Not used
 - **C-SOAP**—Not used
 - **DIAMETER**—Connects the Net-Net SBC to the policy-server
8. **address**—Set this parameter to the IP address of the CLF.
9. **port**—Set this parameter to the port which the CLF uses for Diameter transactions. Port 3868 is the default Diameter port. (The default value is **80**.) The valid range is:
 - Minimum—0
 - Maximum—65535

10. **realm**—Set this parameter to the realm where the CLF exists.
11. **permit-conn-down**—Enable or disable the Net-Net SBC’s ability to permit calls if there is no connection to the external policy server. The default value is **disabled**. The valid values are:
 - enabled | disabled
12. **product-name**—Enter text string that describes the vendor-assigned name for the CLF. This parameter is required.
13. **application-id**—Enter a numeric application ID that describes the interface used to communicate with the RACF. The default value is zero (0). The valid range is:
 - Minimum—0
 - Maximum—999999999
14. **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
15. **watchdog-ka-timer**—Enter the interval in seconds used to determine a DWA response timeout.
16. **num-connections**—Set this parameter to the number of connections the Net-Net SBC will create with the CLF. The default value is 1. The valid range is:
 - Minimum—0
 - Maximum—65535
17. **reserve-incomplete**—Set this parameter to **enabled** if you want the Net-Net SBC to send a message to the CLF that does not include the endpoint’s true port number. A value of 0 will be used for the port number. The default value is **enabled**. The valid values are:
 - enabled | disabled
18. Save your work using the ACCLI **done** command.

```
ACMEPACKET(ext-pol i cy-server)# show
ext-pol i cy-server
      name          CLF
      state         enabled
      operati on-type  admi ssi on-control
      protocol       DI AMETER
      address        192. 168. 115. 10
      port           3869
      real m         aaaaaaa
      permi t-conn-down  di sabl ed
      product-name   CLFAppl i cation
      appl i cati on-id  16777231
      framed-i p-addr-encodi ng  octet -string
      num-connecti ons  500
      reserve-i ncomple te  enabled
```

CLF Debugging

A new argument has been added to the show command for viewing CLF statistics. From the user prompt, type **show <space> ext-clf-svr <return>**.

```

ACMEPACKET# show ext-clf-svr
14:17:14-114
EBM Status          -- Period -- ----- Lifetime -----
                    Active   High    Total     Total  PerMax   High
Client Trans        0        0      0        0      0       0
Server Trans        0        0      0        0      0       0
Sockets             0        0      0        0      0       0
Connections         0        0      0        0      0       0

----- Lifetime -----
Recent      Total  PerMax
CLF Requests  0      0      0
CLF Admits   0      0      0
CLF Errors   0      0      0
CLF Rejects  0      0      0
CLF Expires  0      0      0
CLFD Errors  0      0      0

```

You can also refer to the `log.ebmd` log file located in the `/ramdrv/logs/` directory on the Net-Net SBC. This file must be retrieved via FTP or SFTP.

Diameter e2

In prior releases, the Net-Net SBC's Diameter CLF and RACF interfaces ignore Experimental-Result attribute value pairs (AVPs) if not accompanied by the Result-Code AVP in a Diameter message. With Net-Net Release C5.1, the Net-Net SBC now accepts, parses, and processes the Experimental-Result AVP (which indicates that an error has occurred) whether accompanied by the Result-Code AVP or not. When a CLF or 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: CLF

This section explains how the Net-Net SBC's Diameter CLF interface handles the Experimental-Results AVP and the Result-Code AVP.

CLF Experimental Result Handling

When the Diameter CLF interface receives a User-Data-Answer (UDA) 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 UDA message—either success or failure:

- Success—In the case where the enumerated value indicates success, the Net-Net SBC's Diameter CLF interface internally notifies SIP signaling of its status.
- Failure—In the case where the enumerated value indicates failure (or non-success), the Diameter CLF interface communicates internally with the Net-Net SBC's SIP signaling application to instruct it to use the configured default string in the P-Access-Network-Info header when the value received is either: `DIAMETER_ERROR_USER_UNKNOWN` or `DIAMETER_USER_DATA_NOT_AVAILABLE`.

The Net-Net SBC forwards the Register using the default location string (or without the PANI header if no location string is configured) when the value received is `DIAMETER_UNABLE_TO_COMPLY`.

For all other such result codes, the Register is rejected.

CLF Result Code Handling

When the Diameter CLF interface receives a User-Data-Answer (UDA) message that contains the Result-Code AVP, it determines and performs actions based on the AVP's enumerated value that indicates success or failure:

- Success—In the case where the enumerated value indicates success, the Net-Net SBC's Diameter CLF interface internally notifies SIP signaling of its status.
- Failure—In the case where the enumerated value indicates failure (or non-success), the Diameter CLF interface communicates internally with the Net-Net SBC's SIP signaling application to instruct it to use the configured default string in the P-Access-Network-Info header when the value received is either: DIAMETER_UNABLE_TO_COMPLY. Other failure values will be treated as absolute, and requests will be rejected on that basis.

How It Works: RACF Experimental Result Handling

This section explains how the Net-Net SBC's Diameter RACF interface handles the Experimental-Results AVP.

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.

About Realms and e2 Enhancements

This section describes how the Net-Net SBC treats payload format strings for destination, origin, and host realms.

Destination Realms

The Diameter CLF and 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
<code><user>@<realm></code>	<ul style="list-style-type: none"> • <code><user></code>—IP address of the endpoint initiating the call with the Net-Net SBC • <code><realm></code>—Name of the realm on which the Net-Net SBC received the INVITE from a user
<code><user></code>	<ul style="list-style-type: none"> • <code><user></code>—IP address of the endpoint initiating the call with the Net-Net SBC
<code><realm></code>	<ul style="list-style-type: none"> • <code><realm></code>—Name of the realm on which the Net-Net SBC received the INVITE from a user

When either the Diameter CLF or 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 CLF and RACF interfaces 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

This section shows you how to set the format to use for Destination-Realm AVPs and how to configure a domain name suffix for Origin-Realm and Origin-Host AVPs.

Setting the Destination Realm AVP Format

To set the format to use for the payload string in the Destination-Realm AVP:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.

```
ACMEPACKET(config)# media-manager
ACMEPACKET(media-manager)#
```
3. Type **ext-policy-server** and press <Enter>.

```
ACMEPACKET(media-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
```
4. **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
5. Save and activate your configuration.

Setting the Domain Name Suffix for Origin-Realm and Origin-Host AVPs

To set the suffix to use for Origin-Realm and Origin-Host AVPs that have domain names as payload strings:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.

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

- ```
ACMEPACKET(medi a-manager)#
3. Type ext-policy-server and press <Enter>.
ACMEPACKET(medi a-manager)# ext-policy-server
ACMEPACKET(ext-policy-server)#
4. 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.
5. Save and activate your configuration.
```

## Optional AVP Support

For both RACF and Connectivity Session Location and Repository Function (CLF) functionality, the Net-Net SBC can now 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.

## About the 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 maps the media types and 32-bit unsigned integers.

| m= <media><br>incoming SDP flow | Transport Class AVP Value |
|---------------------------------|---------------------------|
| m=audio                         | 1                         |
| m=video                         | 2                         |

| <b>m= &lt;media&gt;<br/>incoming SDP flow</b> | <b>Transport Class AVP Value</b> |
|-----------------------------------------------|----------------------------------|
| m=application                                 | 3                                |
| m=data                                        | 4                                |
| m=control                                     | 5                                |
| m=image                                       | 6                                |

## 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# config terminal
ACMEPACKET(config)#
```
2. Type **media-manager** and press <Enter>.  

```
ACMEPACKET(config)# 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. Save and activate your configuration.

## Configuring Diameter STR Timeouts

When a call ends, the Net-Net SBC alerts the RACF by sending it a Diameter Session Termination Request (STR) message. You can enable the Net-Net SBC to resend STR messages at the application layer if the STR messages time out. A new option **STR-retry=x** has been created that allows you to configure the number of times the STR messages are resent.

## ACLI Instructions and Examples

### To configure DIAMETER request timeouts:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#

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

```
ACMEPACKET(config)# media-manager
ACMEPACKET(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(medi a-manager)# ext-pol i cy-server
ACMEPACKET(ext-pol i cy-server)#[/pre]

```
4. Set the options parameter by typing **options**, a <Space>, the option name **STR-retry=x** with a “plus” sign in front of it. Then press <Enter>.  

```
ACMEPACKET(ext-pol i cy-server)# options +STR-retry=x
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. Save and activate your configuration.

## Diameter Destination Realm AVP

As of S-C6.2.0, 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 to retain the previous 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        |

## ACLI Instructions and Examples

To enable the Net-Net SBC to send a PS the source realm of the incoming SIP message:

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

```
ACMEPACKET# config terminal
```
2. Type **media-manager** and press <Enter>.  

```
ACMEPACKET(configure)# medi a-manager
ACMEPACKET(medi a-manager)#[/pre]

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

```
ACMEPACKET(medi a-manager)# ext-pol i cy-server
ACMEPACKET(ext-pol i cy-server)#[/pre]

```

If you are enabling this feature on a pre-existing configuration element, then you must select (using the ACLI **select** command) the external policy server that you want to edit.

4. Set the options parameter by typing **options**, a <Space>, the option name **include-gua** with a “plus” sign in front of it. Then press <Enter>.

```
ACMEPACKET(ext-pol i cy-server)# options +include-gua
```

If you type options and then the option value without the “+” 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 “+” sign as shown in the previous example.

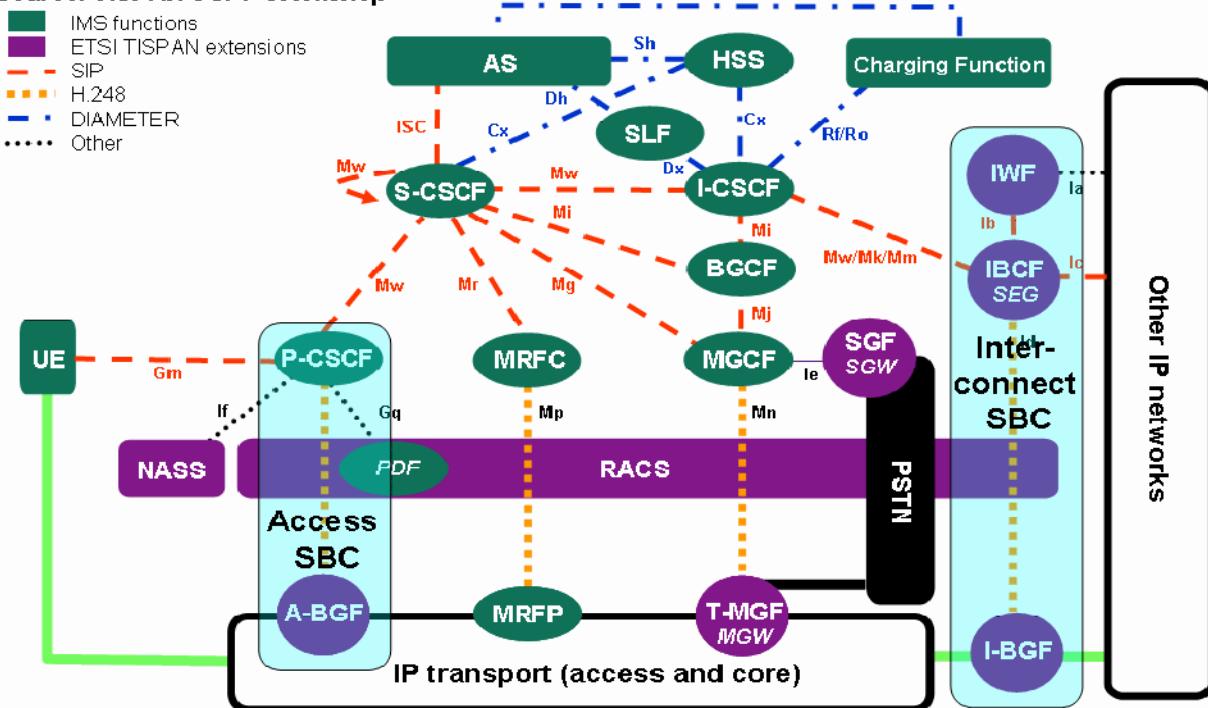
5. Save and activate your configuration.



## Net-Net SBC 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 SBC 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 SBC Access Border Functions

- Proxy CSCF (P-CSCF)
- Access/Core Border Gateway Function (A/C-BGF)
- RACF AF and SPDF functions

### Net-Net SBC Interconnect Border Functions

- Interconnect Border Control Function (I-BCF)
- Interworking Function (IWF)
- Interconnect Border Gateway Function (I-BGF)

## IMS Access Border Functions

---

The Net-Net SBC 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 SBC performs the following functions as the Access SBC:

### P-CSCF Functions

The Net-Net SBC 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 SBC 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 SBC generates real-time accounting records via RADIUS.
- Authorization of bearer resources and QoS management—With integrated BGF capabilities, the Net-Net SBC allocates bearer resources (NAPT flows) and applies QoS policies (including packet marking) based on local policies and/or policies acquired via interaction with the A-RACF (PDF).
- Interaction with the A-RACF (PDF) for session-based policy enforcement and admission control—The Net-Net SBC PDF interface options include COPS and SOAP/XML.
- Traffic Policing—Traffic is policed at the session and media/transport layer. At the signaling layer, the Net-Net SBC 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

### A-BGF Functions

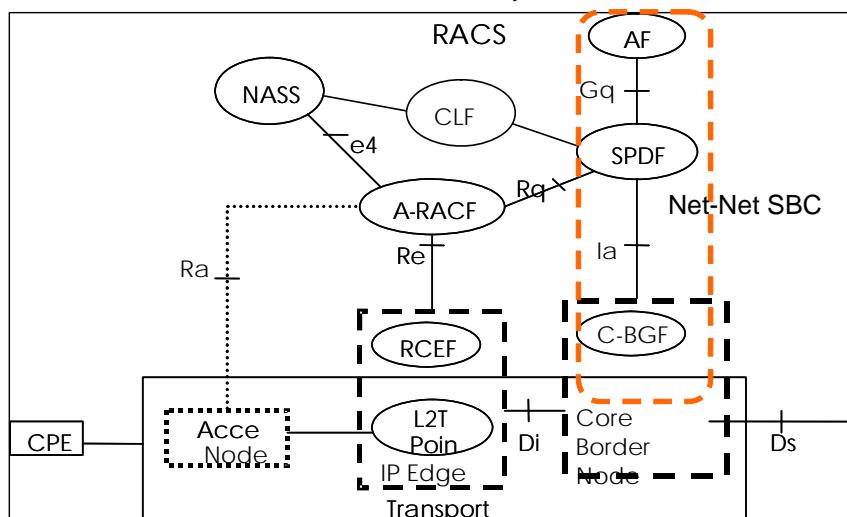
The Net-Net SBC performs the following IMS BGF functions:

- Opening and closing gates/packet filtering—The Net-Net SBC 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 SBC 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 SBC.
- Hosted NAT traversal—The Net-Net SBC supports HNT function that allows media flow traversal through the CPE firewall/NAT without upgrading the CPE equipment. The Net-Net SBC 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 SBC'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 SBC 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.

## Resource and Admission Control (RACS) Functions

The figure below illustrates the mapping of Net-Net SBC functions to the RACS functional model. In this model, the Net-Net SBC incorporates the Application Function (in the case of IMS this is the P-CSCF function), the SPDF (Service Policy Decision Function) and the Core Border Gateway function.



The Net-Net SBC, acting as the SPDF, interfaces with the PDF (A-RACF policy decision function) for resource authorization and admission control on a call-by-call basis. COPS is the supported PDF interface.

## IMS Interconnect Border Functions

---

The Net-Net SBC is deployed at IP interconnect points between service providers to deliver the functions defined in the TISPAN architecture as the I-BCF, IWF and I-BGF. The Net-Net SBC performs the following functions as the Interconnect border SBC:

### **Interworking Function (IWF)**

- Interworking SIP profiles and other protocols (e.g. H.323)

### **Interconnect Border Control Function (I-BCF)**

- Interaction with I-BGF (including NAPT and firewall functions)
- Insertion of the IWF when appropriate
- 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
- Lawful Intercept

## IMS Path and Service Route Header Support

---

The Net-Net SBC supports the Path header and the Service-Route header used in the registration phase of a SIP transaction. The Net-Net SBC 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 SBC. This is how the Net-Net SBC supports RFC 3608 and RFC 3327.

### **Path Header**

When a UE registers to an S-CSCF, the Net-Net SBC 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 SBC, through a specific series of proxies. This route vector is saved in the Net-Net SBC'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 SBC, 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 SBC 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 SBC 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 SBC) 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 SBC 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 SBC 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.

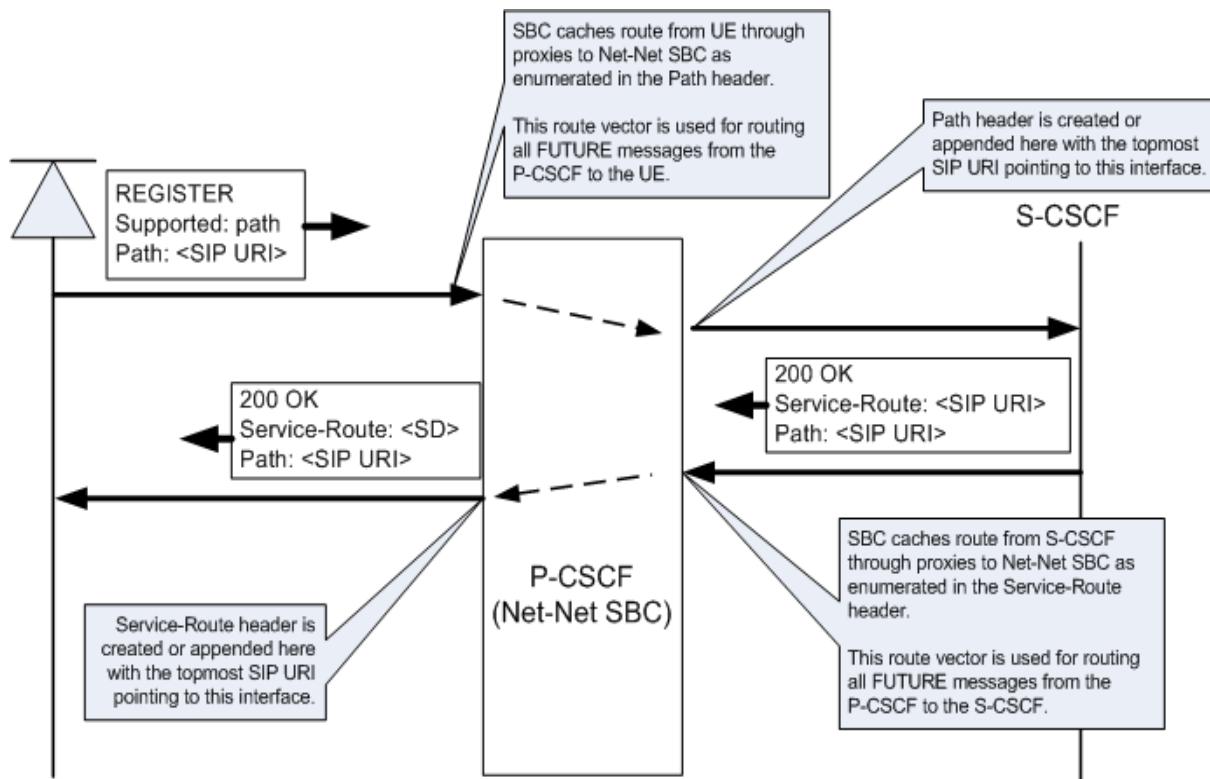
As an exception, you may wish for the Net-Net SBC to not use the Service-Route header to route subsequent Register requests. Note in the configuration section the way to disable Service-Route header routing.

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 SBC. 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 SBC, 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 SBC.

## **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 SBC, you can use the SIP Header Manipulation function.



## Configuring Path and Service Route Headers

This section explains how to configure Path and Service Route headers using the ACLI and the Net-Net EMS.

### ACLI Instructions and Examples

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# config terminal**
2. Type **session-router** and press <Enter> to access the session-level configuration elements.  
**ACMEPACKET(config)# 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. **sip-ims-feature**— Enable IMS functionality on this SIP interface. The default value is **disabled**. Valid values are:
  - enabled | disabled

```
ACMEPACKET(sip-interface)# sip-ims-feature enabled
```

This completes enabling IMS for a given SIP interface.

If you wish to disable subsequent routing of Register messages via the Service-Route header, type **route-register-no-service-route** and press <Enter>.
6. Save your work using the ACLI **done** command.

## IMS Support for Private Header Extensions for 3GPP

---

As part of its RFC 3455 support, the Net-Net SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC 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 SBC will use the configured default P-Asserted-Identity found on the sourcing session agent. This feature works with both SIP and H.323 session agents.
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.

## Configuring P-Asserted-Identity Header for Session Agents

### 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 earlier 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# config terminal
```

2. Type **session-router** and press <Enter> to access the session-level configuration elements.  

```
ACMEPACKET(configure)# sessi on-router
ACMEPACKET(sessi on-router)#

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

```
ACMEPACKET(sessi on-router)# sessi on-agent
ACMEPACKET(sessi on-agent)#

```
4. Type **select** and the number of the pre-configured session agent you want to configure.  

```
ACMEPACKET(sessi on-agent)# select 1
```
5. Type **p-asserted-id** <space> <URI to use when no P-Asserted-ID has been created> and press <Enter>. This completes the configuration.  

```
ACMEPACKET(sessi on-agent)# p-asserted-id sIp: name@acmepacket.com
```
6. Save your work using the ACCLI **done** command.

## P-Called-Party-ID Header

The Net-Net SBC transparently passes the P-Called-Party-ID header between the S-CSCF and a UA.

## IMS Charging Headers

The Net-Net SBC 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 now 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.

Charging headers are inserted, deleted, or ignored for request messages. They are forwarded through the Net-Net SBC unmodified when embedded in response messages. If you wish to modify the charging headers in a response message, you must use the Net-Net SBC's header manipulation feature as a general solution.

## P-Charging-Vector

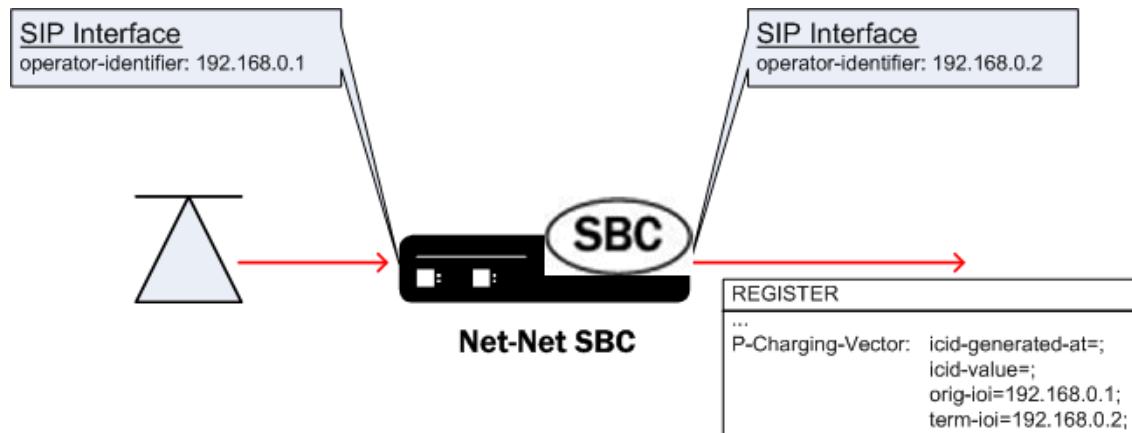
You can configure the Net-Net SBC to processes the P-Charging Vector header in three different ways.

- If a P-Charging-vector header is present in an incoming SIP request, the Net-Net SBC can pass the header untouched, as part of the full SIP message that is forwarded out of an egress interface.
- If a P-Charging-vector header is present in an incoming SIP request, the Net-Net SBC can delete the header and forward the full SIP message out of an egress interface.
- If an incoming SIP request does not contain a P-charging-vector header, the Net-Net SBC can create and insert the header and forward the full SIP message out of an egress interface. Likewise, if an incoming SIP request contains an existing P-Charging-Vector header, the Net-Net SBC can overwrite this header with the values generated internally.

The P-Charging-Vector header is composed of four parameters: icid-value, icid-gen-addr, orig-ioi, term-ioi. See RFC 3455, Section 4.6 for more information.

1. The Net-Net SBC constructs the icid-value in the following format:  
string2@string1 where:  
String 1 is the IP address of the egress SIP interface.  
String 2 is a unique string value created by the Net-Net SBC and based on the realm, local IP port, time, and a sequence number.
2. The icid-gen-addr parameter's value is the IP address of the egress SIP interface. This value is generated by the Net-Net SBC.
3. The orig-ioi parameter's value is set manually using the operator-identifier field located in the SIP interface configuration element.
4. The term-ioi parameter's value is set manually using the operator-identifier field located in the SIP interface configuration element.

You configure charging vector handling on the Net-Net SBC interface that receives the SIP request by turning on the switches that enable charging vector processing on the ingress interface for the call. Based on the direction of the call, the Net-Net SBC will insert the operator-identifier configuration 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-Vector Header Example

```

P-Charging-Vector: icid-
value=1ate6g46n1823s8719ck3ps6gbt46m5d3bci3po5hhdg3n86g1csi047g9c43@1
92.168.0.2;
icid-generated-at=192.168.0.2;
orig-ioi=192.168.0.1;
term-ioi=192.168.0.2;

```

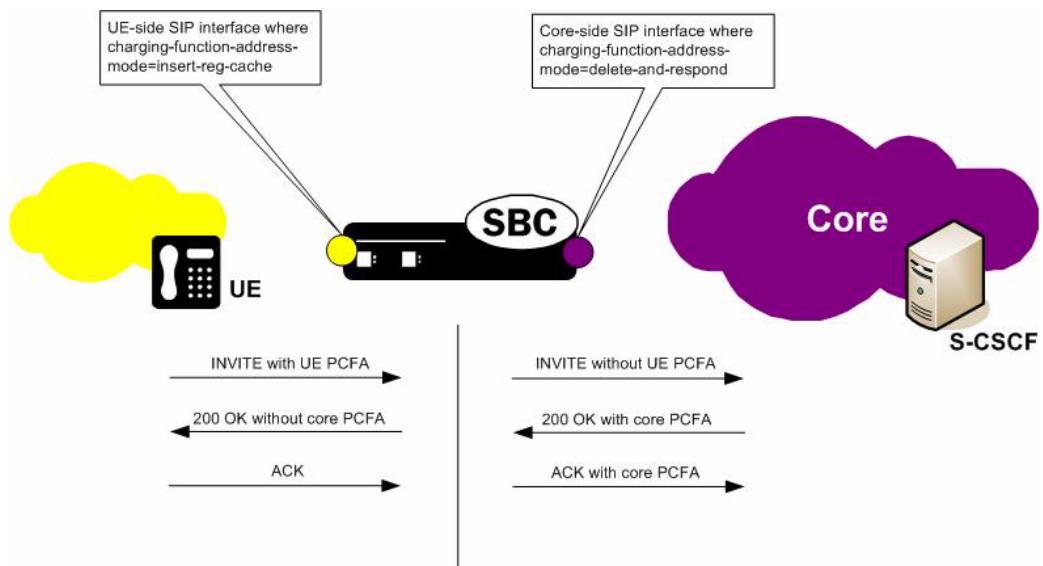
### P-Charging-Function-Address

The P-Charging-Function-Address header is composed of two configurable parameters: ccf, ecf. You can configure the Net-Net SBC to processes the P-Charging-Function-Address header in three different ways.

- If a P-Charging-Function-Address header is present in an incoming SIP request, the Net-Net SBC can be set to pass the header, untouched, as the full SIP request is forwarded out of an egress interface.

- If a P-Charging-Function-Address header is present in an incoming SIP request, the Net-Net SBC can be set to delete the header and forward the SIP request out of an egress interface.
- If an incoming SIP request does not contain a P-Charging-Function-Address header, the Net-Net SBC can be set to create and insert the header and forward the SIP message out of an egress interface.
- If an incoming SIP request contains a P-Charging-Function-Address header, and the Net-Net SBC is set to insert a configured P-Charging-Function-Address header, the new parameters will be appended before the existing parameters in the header. The Net-Net SBC will then forward the SIP request out of an egress interface.

In addition, the Net-Net SBC can also be configured to perform insertion and caching for the PCFA header in dialog-creating or stand-alone messages. The following diagram illustrates how this works:



For this scenario, there are two main functions, PCFA insertion and PCFA caching:

- PCFA insertion—Using the insert-reg-cache and delete-and-respond configuration values, the Net-Net SBC adds the PCFA to all SIP requests and to the response on the S-CPCF facing the SIP interface. However, only dialog-creating and standalone requests, and responses to each of those, update the Net-Net SBC and accounting information. Such requests do not have a To tag, and responses do not appear in established dialogs. The Net-Net SBC inserts the PCFA into provisional (1XX) and success (2XX) responses, with the exception of the 100 Trying response.

You can use SIP header manipulation rules (HMR) to remove any unwanted headers.

- PCFA caching—When you use either of the insert-reg-cache and delete-and-respond configuration values, the Net-Net SBC uses the latest cached copy of a PCFA header to insert into requests and responses. The Net-Net SBC does not cache any PCFA headers it receives on SIP interfaces using the none, insert, or insert-reg-cache modes because this type of SIP interface faces the UE making its replacement headers ones from the core.

Though there can be various sources for the latest cached copy, the PCFA header received as part of a dialog-creating or standalone request has highest precedence. This PCFA header is then stored as the latest cached value for that dialog. That is, for each specific dialog, the Net-Net SBC the PCFA is cached separately so it can add the most specific PCFA to the message—and is added to any message for the dialog.

When there is no cache PCFA for a specific dialog, the Net-Net SBC uses the registration cache entry as the latest cached copy. And when there is no entry in the registration, the PCFA uses the **ccf-address** and **ecf-address** values from the SIP interface.

The latest cached copy or the **ccf-address** is the value reported in the RADIUS VSA Acme-Session-Charging-Function-Address; this VSA is used for both of the new modes. Note that only the **ccf-address** is reported in RADIUS records; the **ecf-address** is not.

### P-Charging-Function-Address Header Example

P-Charging-Function-Address: ccf=192.168.0.20 ; ecf=192.168.0.21;

### RADIUS Accounting of Charging Headers

When the Net-Net SBC 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 iid-value is saved to the P-Charging-Vector attribute in the radius record. If the Net-Net SBC does not create a P-Charging-Vector header, but it receives a SIP message that already has the P-Charging-Vector header with an iid-value, the existing iid-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 SBC creates the P-Charging-Function-Address, the CCF value it inserts into the header is saved to the radius record. If the Net-Net SBC 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     |

### Configuring P-Charging-Vector Processing for SIP Interfaces

This section explains how to configure P-Charging-Vector processing using the ACLI.

### ACLI Instructions and Examples

P-Charging-Vector header handling is enabled in the SIP interface.

To configure P-Charging-Vector processing in 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 support to an existing SIP interface, then you will need to select the interface you want to edit using the ACLI **select** command.

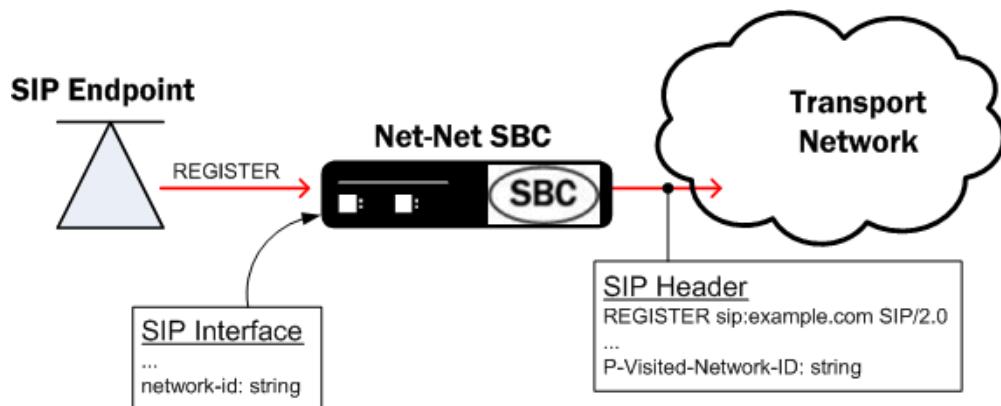
```
ACMEPACKET(sip-interface)# select first-sip-interface
```
4. **charging-vector-mode**—Sets the Net-Net SBC to create the P-Charging-Vector header. The default value is **pass**. The valid values are:
  - **none**—Pass this header unchanged (do not include icid-value in accounting records)
  - **pass**—Pass this header unchanged (include icid-value in accounting records)
  - **delete**—Delete this header
  - **insert**—Create this header
5. **operator-identifier**—Set the operator identifier value to be inserted into a P-Charging-Vector header. The direction of the call determines whether this value is inserted into the orig-roi or the term-roi parameter in the P-Charging-Vector header. This string MUST begin with an alphabetical character.
6. **charging-function-address-mode**—Set the charging function address mode you want to use. The default value is **pass**. The valid values are:
  - **none**—Pass the Charging-Function-Address header received in an incoming SIP message untouched as the message is forwarded out of the Net-Net SBC, but does not include icid-value in accounting records.
  - **pass**—Pass the Charging-Function-Address header received in an incoming SIP message untouched as the message is forwarded out of the Net-Net SBC, includes icid-value in accounting records.
  - **delete**—Delete the Charging-Function-Address header received in an incoming SIP message before it is forwarded out of the Net-Net SBC.
  - **insert**—Insert the Charging-Function-Address header in an incoming SIP message that does not contain the Charging-Function-Address header. If the incoming message contains the Charging-Function-Address header, the Net-Net SBC will overwrite the Charging-Function-Address header with its values. This option always uses the **ccf-address** and **ecf-address** static values.
  - **insert-reg-cache**—To be configured on the SIP interface facing the UE, configures the Net-Net SBC to replace the PCFA with the most recently cached value rather than the **ccf-address** and **ecf-address** you set to be static in your configuration. The cached values come from one of the following that the Net-Net SBC has received most recently: request, response, registration, or local configuration.
  - **delete-and-respond**—To be configured on the SIP interface facing the S-CPCF, configures the Net-Net SBC to strip out the latest cached PCFA from the core side. The Net-Net SBC then remembers this PCFA and uses it in communications sent to the core.

Note that the default settings for this parameter and for **charging-vector-mode** are **pass** for new SIP interface configurations. If you are upgrading and there are pre-existing SIP interfaces in your configuration, the defaults become **none**.

7. **ccf-address**—Set the CCF address value that will be inserted into the P-Charging-Function-Address header.
8. **ecf-address**—Set the ECF address value that will be inserted into the P-Charging-Function-Address header.
9. Save your work using the ACLI **done** command.

## P-Visited-Network-ID Header

The Net-Net SBC'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 SBC 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.

## Configuring P-Visited-Network-ID Header Handling for SIP Interfaces

### ACLI Instructions and Examples

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>.   
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 **network-id** <space> <network ID string> and press <Enter>. This completes the configuration of the P-Visited-Network-ID string for a given SIP interface.  

```
ACMEPACKET(sip-interface)# network-id examplenetworkid

```
6. Save your work using the ACLI **done** command.

## Surrogate Registration

---

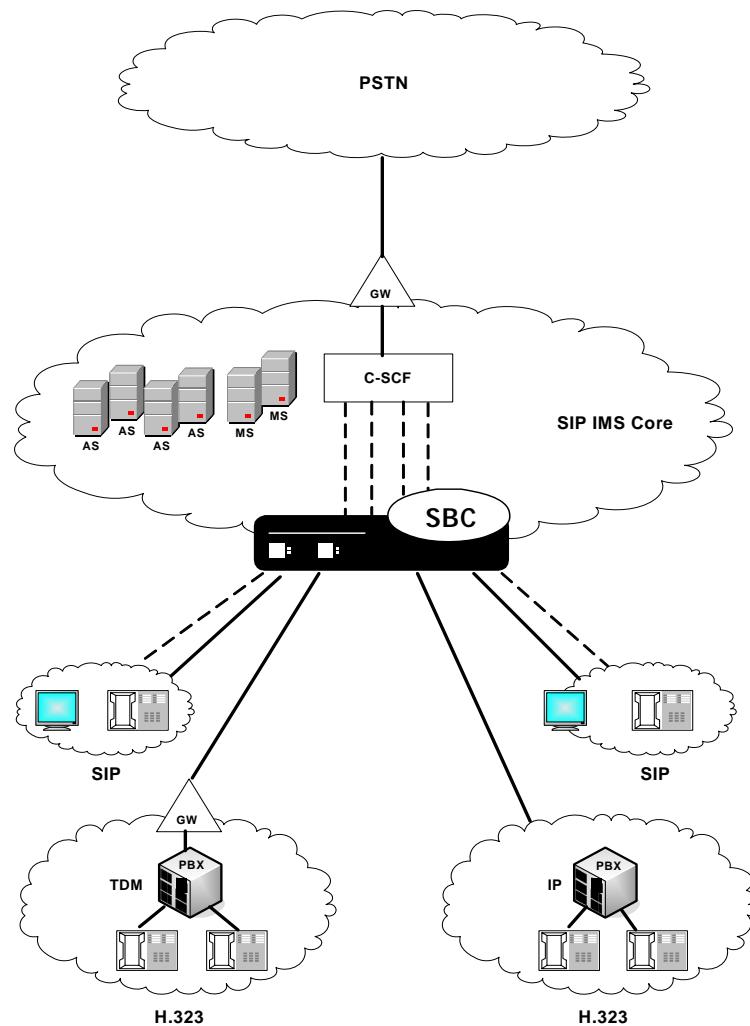
The Net-Net SBC surrogate registration feature lets the Net-Net SBC explicitly register on behalf of a Internet Protocol Private Branch Exchange (IP-PBX). After you configure a surrogate agent, the Net-Net SBC periodically generates a REGISTER request and authenticates itself using a locally configured 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-PBXes 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 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 or H.323 IP-PBXes. The Net-Net SBC performs SIP registration on behalf of the IP-PBX and IADs.



## How it Works

The Net-Net SBC registers on behalf of the IP-PBXes and then stores the associated URIs returned by the Serving Call Session Control Function (S-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 might only be able to 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. 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.

### H.323

If the session agent or the session agent group configured for the customer-next-hop parameter references an H.323 device, the Net-Net SBC sends the INVITE to its interworking task. If a session agent group is being used, the parameter containing the session agent group name is added to the Request-URI. The host portion of the Request-URI will point to the interworking IP address and the port is changed to 1720.

If a session agent is used, the Net-Net SBC uses it to route the call properly to the interworking task to take care of the H.323 call setup.

## 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 same as the one configured for the register-host parameter.

With an H.323 IP-PBX, when the Net-Net SBC receives an INVITE from the interworking task it looks to see if the call is coming from a session agent. If it is, it looks to see if this session agent has a surrogate agent configured. If it does, the Net-Net SBC changes the host portion of the From header to match the registration entry stored in the registration cache.

After the corresponding registration Service-Router entry is found, the Net-Net SBC 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.l

### To configure the surrogate agent:

1. In Superuser mode, type **configure terminal** and press <Enter>.  
**ACMEPACKET# config terminal**
2. Type **session-router** and press <Enter> to access the system-level configuration elements.  
**ACMEPACKET(config)# 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 (Address of Record).
6. **state**—Set the state of the surrogate agent to indicate whether the surrogate agent is used by the application. The default value is **enabled**. The valid values are:
  - enabled | disabled
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 proxy 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. The default value is **600,000** (1 week). The valid range is:
  - Minimum—0
  - Maximum—999999999
15. **replace-contact**—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. The default value is **disabled**. The valid values are:
  - enabled | disabled
16. **route-to-registrar**—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. The default value is **enabled**. The valid values are:
  - enabled | disabled

17. **aor-count**—Enter the number of registrations to do on behalf of this IP-PBX. If you enter a value greater than 1, 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. The default value is 1. The valid range is:
  - Minimum—0
  - Maximum—999999999
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. Save and activate your configuration.

## Example

The following example shows the surrogate agent configuration.

```
surrogate-agent
 regi ster-host acmepacket.com
 regi ster-user 234567
 state enabled
 real m-i d public
 descri pti on
 customer-host acmepacket.com
 customer-next-hop 111. 222. 333. 444
 regi ster-contact-host 111. 222. 5. 678
 regi ster-contact-user eng
 password
 regi ster-expi res 600000
 repl ace-contact disabled
 route-to-registrar enabled
 aor-count 1
 opti ons
 auth-user
 I ast-modi fi ed-date 2006-05-04 16: 01: 35
```

## SIP Surrogate Registration Enhancements

For IMS-E networks, enhancements to the Net-Net SBC's SIP surrogate registration capabilities enable it to register a series of endpoints on behalf of a set of devices that are unable to register themselves. In addition, the Net-Net SBC retries failed registrations, prevents authentication loops, and sends an SNMP trap for failed retransmissions. The automatic incrementing of register-user and register-contact-user values are also now more flexible.

## Without Enhancements

Without the enhancements configured, the Net-Net SBC's surrogate agent performs a series of registrations based on count when the system boots or when its configuration changes. It only attempts to register each user once. Although the surrogate agent uses the same retry mechanism used for SIP client transactions, it does not attempt further if it receives a failure response until the entry expires. When it 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.

## With Enhancements

With the enhancements configured, the Net-Net SBC supports:

- Registration retry—You can configure the surrogate agent to retry registration when 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—Authentication loops can occur in previous releases when the Net-Net SBC resends a registration request with authentication in response to 401, 403, or 407 responses (indicating, for example, that there might be a password error). Using the new enhancements, 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 enhancements—Now, the automatic increment 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 also 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.

## Configuring the Retry Mechanism

### To set the retry mechanism:

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

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

```
ACMEPACKET(config)# session-router
ACMEPACKET(session-router)#
```
3. Type **surrogate-agent** and press <Enter>.  

```
ACMEPACKET(session-router)# surrogate-agent
ACMEPACKET(surrogate-agent)#
```
4. **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 value is 3. The valid range is:
  - Minimum—0
  - Maximum—10
5. **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 value is 300. The valid range is:
  - Minimum—10

- Maximum—3600
6. Save and activate your configuration.

## Configuring the Count Start

### To set the value where automatic incrementing will start:

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 **surrogate-agent** and press <Enter>.  

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

```
4. **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 default value is 1. The valid range is:
  - Minimum—0
  - Maximum—999999999

## IMS Implicit Service Route

---

The Net-Net SBC provides implicit service route support in situations where it is deployed between user equipment (UE) and the P-CSCF, and where the IMS core network does not support the Service-Route header.

When this feature is enabled, the Net-Net sends requests to the P-CSCF and does not include the Service-Route header received in the 200 OK response (to a REGISTER message) as Route headers in subsequent requests. The Net-Net SBC also includes a Route header of the P-CSCF address in subsequent requests, and includes the loose-route parameter in the Route header. Because inclusion of the loose-route parameter is not needed in all cases, you can set this feature to “strict” in the SIP interface configuration.

Recent enhancements address the following 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 received from the REGISTER. Therefore, you must use SIP HMR to remove the Service-Route headers from the response, while having no mechanism to replace 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-registrations-no-service-route** sets the behavior for using the Service-Route header in the REGISTER request. 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.

## How It Works

When implicit service route support is enabled, the Net-Net SBC stores the Service Route URIs from the Service-Route headers that are included in 200 OK responses to REGISTER messages. The Service Route URIs are included in the Route headers in subsequent Request messages, except for REGISTER messages.

The Net-Net SBC also supports the ability to keep the loose-route parameter from being included in the implicit Route URI that the Net-Net SBC generates and includes as a Route header in the Request messages.

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 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(config)# **session-router**
3. Type **sip-interface** and press <Enter>.  
ACMEPACKET(session-router)# **sip-interface**
4. **implicit-service-route**—To enable implicit service route support, change this parameter from **disabled** to **enabled**. The default value is **disabled**. Supported values are:  
 absent | disabled (the default) | enabled | replace | strict
  - **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.
  - **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.
  - **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.
  - **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.

Save and activate your configuration.

**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-registrars-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-registrars-no-service-route** option. If the system finds the option, then it translates the value configured for the option like this:

| Old route-register-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.

## IMS Charging Vector Mode Adaptation

This adaptation to the Net-Net SBC's IMS functionality provides the ability to remove the P-Charging-Vector from incoming requests for a session and store it. Then the Net-Net SBC inserts it into outbound responses related to that session in a P-Charging-Vector header.

### ACLI Instructions and Examples

Typically, the ACLI **charging-vector-mode** parameter is set to **delete-and-respond** (which supports removing and storing the P-Charging-Vector for later insertion in outbound response) on the core, trusted interface. On the access, untrusted side, this same parameter is set to **insert**.

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

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

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```
3. Type **sip-interface** and press <Enter>. If you are adding this feature to a pre-existing SIP interface, you need to select and edit that configuration.  

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```
4. **charging-vector-mode**—Change this parameter to **delete-and-respond** to remove the P-Charging-Vector from incoming requests for a session and store it. Then the Net-Net SBC inserts it into outbound responses related to that session in a P-Charging-Vector header.
5. Save and activate your configuration.

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

```
ACMEPACKET(config)# 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-AKA

---

The Net-Net SBC supports IP Media Subsystem-Authentication and Key Agreement (IMS-AKA).

Defined in 3GPPR7 (specifications in TS 33.203 and call flows in TS 24.228), IMS-AKA can be used as a framework for authentication and for securing the signaling path between a UE and the Net-Net SBC (when the Net-Net SBC is acting as a P-CSCF or as a B2BUA) across the Gm interface.

In addition, the Net-Net SBC’s serving as an IMS-AKA termination point is valuable because it allows IMS-AKA use behind by multiple endpoints sitting behind a NAT device. IMS-AKA support also works when there are no NAT devices between endpoints and the Net-Net SBC acting as a P-CSCF, and when the Net-Net SBC sits behind a third-party P-CSCF. In addition, you can use IMS-AKA when the endpoint uses SIP UDP.

## Requirements

IMS-AKA use assumes that you have installed the appropriate IPSec and SSM modules on your Net-Net SBC, or that it has come from Acme Packet with those modules pre-installed. IMS-AKA will not work without this hardware.

In addition, your configuration must have SIP registration caching enabled.

## Monitoring

The ACLI **show sipd endpoint-ip** command is updated to show the IMS-AKA parameters corresponding to each endpoint. The display shows the algorithms used, the ports used, and the security parameter indexes (SPIs) used.

In addition, the **show sa stats** command now shows the security associations information for IMS-AKA.

## ACLI Instructions and Examples

### Setting Up an IMS-AKA Profile

You enable IMS-AKA by configuring the following:

- An IMS-AKA profile
- Certain parameters in the global IPSec configuration
- Certain parameters in the SIP interface, and in the SIP interface's SIP port

An IMS-AKA profile establishes the client and server ports to be protected, and it defines lists of encryption and authentication algorithms the profile supports. You can configure multiple IMS-AKA profiles, which are uniquely identified by their names.

You apply an IMS-AKA profile to a SIP port configuration using the name.

#### To configure an IMS-AKA profile:

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

```
ACMEPACKET# config terminal
ACMEPACKET(config)#
```
2. Type **security** and press <Enter>.  

```
ACMEPACKET(config)# security
ACMEPACKET(security)#
```
3. Type **ims-aka-profile** and press <Enter>.  

```
ACMEPACKET(system)# ims-aka-profile
ACMEPACKET(ims-aka-profile)#
```
4. **name**—Enter the name you want to give this IMS-AKA profile. This is the value you will use to apply the profile to a SIP port configuration. This parameter is required, and it has no default value.
5. **protected-server-port**—Enter the port number of the protected server port, which is the port on which the Net-Net SBC receives protected messages. The protected server port should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.  

This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.
6. **protected-client-port**—Enter the port number of the protected client port, which is the port on which the Net-Net SBC sends out protected messages. Like the protected server port, the protected client port should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.  

This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.
7. **encr-alg-list**—Enter the list of encryption algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

```
ACMEPACKET(ims-aka-profile)# encr-alg-list "aes-cbc null"
```

This parameter defaults to the following three values: **aes-cbc**, **des-ed3-cbc**, and **null**.

8. **auth-alg-list**—Enter the list of authentication algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

```
ACMEPACKET(ims-aka-profile)# auth-alg-list "hmac-sha-1-96 hmac-md5-96"
```

This parameter defaults to **hmac-sha-1-96**.

## **Setting Up an IPSec Profile for IMS-AKA Use**

Using the global IPSec configuration, you establish the parameters governing system-wide IPSec functions and behavior. This configuration also contains parameters required for IMS-AKA support. The IPSec global configuration is a single instance element, meaning there is one for the whole system.

### **To configure the global IPSec parameters that apply to IMS-AKA:**

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

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

```

2. Type **security** and press <Enter>.

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

```

3. Type **ipsec** and press <Enter>.

```
ACMEPACKET(system)# ipsec
ACMEPACKET(ipsec)#

```

4. Type **ipsec-global-config** and press <Enter>. If you are editing a pre-existing IPsec global configuration, then you need to select the configuration before attempting to edit it.

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

```

5. **red-ipsec-port**—Specify the port on which the Net-Net SBC should listen for redundancy IPSec synchronization messages. The default is 1994, and valid values are in the range from 1025 to 65535.

6. **red-max-trans**—Enter the maximum number of redundancy transactions to retain on the active. The default is 10000, and valid values range up to a 999999999 maximum.

7. **red-sync-start-time**—Enter the time in milliseconds before the system starts to send redundancy synchronization requests. The default is 5000, and valid values range up to a 999999999 maximum.

8. **red-sync-comp-time**—Enter the time in milliseconds to define the timeout for subsequent synchronization requests once redundancy synchronization has completed. The default is 1000, and valid values range up to a 999999999 maximum.

## **Enabling IMS-AKA Support for a SIP Interface**

To enable IMS-AKA for a SIP interface, you must set the **ims-aka-feature** parameter to enabled.

### **To enable IMS-AKA for a SIP interface:**

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

- ```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
2. Type session-router and press <Enter>.
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
3. Type sip-interface and press <Enter>. If you are adding this feature to a pre-existing SIP interface, you need to select and edit that configuration.
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
4. ims-aka-feature—Change this parameter to enabled if you want to use IMS-AKA on this SIP interface. By default, this parameter is disabled.
```

Applying an IMS-AKA Profile to a SIP Port

The final step in setting up IMS-AKA support is to apply an IMS-AKA profile to a SIP port. Enter the **name** value from the IMS-AKA profile you want to apply in the SIP port's **ims-aka-profile** parameter.

To apply an IMS-AKA profile to a SIP 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>.
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
4. Type sip-interface and press <Enter>. If you are adding this feature to a pre-existing SIP port, you need to select and edit that configuration.
ACMEPACKET(session-interface)# sip-ports
ACMEPACKET(sip-port)#
5. ims-aka-profile—Enter the name value for the IMS-AKA profile configuration you want applied to this SIP port. This parameter has no default.
6. Save and activate your configuration.
```

SIP, IMS, P-CSCF: P-Asserted Identity in Responses

In releases earlier than Release S-C6.1.0, the Net-Net SBC—operating as a P-CSCF—removes the P-Preferred-Identity header (if present) on receipt of a 1xx or 2xx response. It also inserts a P-Asserted-Identity header with the value received in the P-Preferred-Identity header.

Release S-C6.1.0 changes this behavior. Now 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 relate 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.

E-CSCF Support

An Emergency Call Session Control Function (E-CSCF) is an IMS core element that aids in routing emergency calls to an appropriate destination, such as a PSAP. E-CSCF functionality can be performed by the Net-Net SBC with appropriate local policy and network management control configuration.

The E-CSCF feature let the Net-Net SBC internally prioritize and route emergency calls to the corresponding Emergency Service Center, based either on the calling party's request URI, or based on location information retrieved from a CLF (Connectivity Location Function) for wireline/TISPAN networks.

By integrating E-CSCF functionality into the P-CSCF (Net-Net SBC), networks can satisfy the common local requirement that certain telephony elements be deployed locally, rather than use single, centralized elements. Functions like the E-CSCF likely fall into this category.

Service URN Support

To enable E-CSCF functionality, the Net-Net SBC can parse service URNs for local policy lookup keys, and as destination identifiers in network management controls (NMC). Ensure that the match-URN is entered correctly as: "urn:service:sos" or "urn:service:sos.type" or the Net-Net SBC will interpret the URN as a hostname. Please see RFC 5031 for more information on compliant URN construction.

E-CSCF Configuration Architecture

There are four elements which comprise and enable E-CSCF support on the Net-Net SBC:

1. CLF Connectivity
2. NMC Emergency Call Control
3. Local Policy
4. Emergency Local Route Table

CLF Connectivity

The Net-Net SBC must be configured with Diameter-based CLF support. This is accomplished by creating an appropriate external policy server configuration. Please see the [Connectivity Location Function \(1012\)](#) section in the External Policy Servers chapter for more information.

When the Net-Net SBC requests authorization from the CLF server, a Line-Identifier AVP which includes a location string is expected to be returned for the call. The returned location string will be used later for an LRT query.

NMC Emergency Call Control

By configuring a Network Management Control (NMC), the Net-Net SBC can flag a call for special priority early after it is received and validated by the system. The **destination identifier** must be configured in the NMC with the service URN of an incoming emergency call. Also, the NMC configuration must have its **next hop** parameter left blank. This lets the Net-Net SBC route the emergency call with local policies.

For example, if **urn:service:sos** is the configured value in the NMC's **destination identifier**, and an INVITE arrives on the Net-Net SBC with **urn:service:sos** in the request URI, the call will be flagged for emergency handling. The next step in call processing is for the INVITE to be evaluated by local policy.

Local Policy

Local policies must be configured to match and then route an incoming emergency call. Once a local policy match is made, the Net-Net SBC looks to the configured policy attributes for where to forward the INVITE. A matching policy attribute's next hop should be configured to point to an emergency LRT that contains specific destinations for emergency calls. In addition, the **elec str lkup** parameter must be set to enabled so the Net-Net SBC will perform an LRT lookup based on the location string returned in the CLF response.

The **eloc str match** parameter identifies the attribute, whose value in the location string will be used as the lookup key in the emergency LRT. For example, if the returned location string is:

```
loc=xxx; noc=yyyy; line-code=zzzz
```

and the **eloc str match** parameter is set to **noc**, then when the Net-Net SBC performs a local policy route search, it will search the LRT for **yyyy**. If the **eloc str match** parameter left empty or if there is no match when **eloc str lkup** is enabled, the entire location string is used as the lookup key.

Emergency LRT

The Net-Net SBC needs to be configured with an emergency LRT to route emergency calls to their destination.

As stated in the previous section, when searching an emergency LRT, any user defined parameter within a Location String may be used as the key to look up next-hop routing information.

LRT files support `<user type = "string">` which enables the Net-Net SBC to perform searches on free form attributes that may appear in the returned location-string. The `<user type = "string">` value for an entry in the emergency LRT should be set to a part or whole value returned in the CLF's location string. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Local Routes>
  <route>
    <user type="string">1234</user>
```

```

<next type="regex">! ^.*$! si p: 911@192.168.200.140:5060! </next>
</route>
<route>
<user type="string">loc=xxx; noc=yyy; line-code=zzzz</user>
<next type="regex">! ^.*$! si p: 911@192.168.1.139:5060! </next>
</route>
</local Routes>
```

Note: Given that the Location String is not a well-defined string, care should be taken when defining and configuring the LRT tables.

Note: LRTs must be individually uploaded to both the active and standby systems in an HA node; LRTs are not automatically replicated across nodes.

CLF Response Failure

If there is no location string in a CLF's response or the CLF rejects the call, the Net-Net SBC uses the **default location string** parameter from the ingress SIP interface to populate the PANI header. The emergency call proceeds normally using this location string's information for emergency LRT lookups.

ACLI Instructions and Examples

This procedure assumes that the Net-Net SBC is configured to communicate with a CLF. In addition, this procedure assumes an the Net-Net SBC is configured and loaded with an appropriate LRT for E-CSCF Use.

To configure an NMC for E-CSCF use (baseline parameters are not mentioned):

1. In Superuser mode, type **configure terminal** and press <Enter>.
ACMEPACKET# config terminal
2. Type **session-router** and press <Enter> to access the signaling-related configurations.
ACMEPACKET(config)# 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**—Enable or disable this network management control rule. The default value is **enabled**. The valid values are:
 - enabled | disabled
6. **type**—Set this parameter to **priority** so that the Net-Net SBC will flag incoming calls with a matching destination identifier as a priority calls.
7. **treatment**—Set this parameter to **divert**.
8. **next-hop**—Leave this parameter blank so that the call's processing will go directly to local policy.
9. **destination-identifier**—Enter the service URN that endpoints in your network include in their request URIs to identify themselves as emergency calls.
10. Save your configuration.

To configure local policy for E-CSCF use (baseline parameters are not mentioned):

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. **to-address**—Set this parameter to the lookup key for matching emergency calls. You can now use a service URN as lookup key criteria.

5. Save your configuration.

To configure policy attributes for E-CSCF use (baseline parameters are not mentioned):

6. 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)#

```

7. **next-hop**—Set this parameter to **lrt:name-of-elrt-file.gz** for this policy attribute to lookup routes in the named lrt file.

8. **eloc-str-lkup**—Set this parameter to **enabled** for the Net-Net SBC to parse the emergency location string, as received in a CLF Line Identifier AVP, for emergency LRT lookup.

9. **eloc-str-match**—Set this parameter to the attribute name found in the location string whose value will be used as a lookup key in the LRT named in the next-hop parameter. Common values include "loc" or "noc".

10. Save and activate your configuration.

Maintenance and Troubleshooting

The **show lrt route-entry** command displays two entries, if the username 1234 has a "string" type and "E164" type entries.

```
ACMEPACKET# show lrt route-entry emergency_lrt 1234
UserName <1234>
User Type= E164
NextHop= !^. *$! sip: 911@192. 168. 200. 139: 5060!
NextHop Type= regexp
UserName <1234>
User Type= string
NextHop= !^. *$! sip: 911@192. 168. 200. 140: 5060!
NextHop Type= regexp
```


Acronym List

General Use Acronyms

3GPP— 3rd-Generation Partnership Project

A

- AAA—Authentication, Authorization, and Accounting
- ACD—Automatic Call Distribution
- ACL—Access Control List
- ACLI—Acme Command Line Interface
- ACP—Acme Control Protocol
- ADMF—ADMInistration Function
- AF—Access Function
- AFID—Access Function Identifier
- AIN—Advanced Intelligent Network
- ALG—Application Layer Gateway
- ALIP—Acme Lawful Intercept Protocol
- AM—Application Manager
- ANI —Automatic Number Identification (ISDN)
- ANSI—American National Standards Institute
- AoR—Address of Record
- AP—Application Protocol
- API—Application Programming Interface
- APN—Access Point Name
- APPN—Advanced Peer-to-Peer Networking
- ARP—Address Resolution Protocol
- ARQ—Admission Request (H.323)
- ASCII—American Standard Code for Information Interchange
- ASIC—Application-Specific Integrated Circuit
- ASN—Abstract Syntax Notation
- ASN.1—Abstract Syntax Notation – 1
- ASP—Application Service Provider, Active Server Pages, Adjunct Service Point
- ASR —Access Service Request
- ATCP—Async TCP

ATCP socket—Refers to a socket used for an async TCP connection.
 ATCP stack—Refers to the separate TCP stack implemented in the ATCP task.
 ATCP task—Refers to the task in the system in which the ATCP stack executes.
 ATM—Asynchronous Transfer Mode
 ATX—Advanced Technology Extended

B

B2BGW—Back-to-Back Gateway
 B2BUA—Back-to-Back User Agent
 BBSRAM—Battery Backup Static Random Access Memory
 BC—Bearer Capability
 BCID—Billing Correlation Identifier
 BER—Basic Encoding Rules
 BGF—Border Gateway Function
 BGP—Border Gateway Protocol
 BHCA—Busy Hour Call Attempts
 BIOS— Basic Input/Output System
 BIS—Bearer-Independent Setup
 BITS—Building Integrated Timing Supply
 B-ICI—Broadband Inter-carrier Interface (ATM)
 B-ISUP—Broadband ISDN User Part
 BNF—(augmented) Backus-Naur Form
 BoD—Bandwidth on Demand
 BoS—Bottom of Stack
 bps—Bits per Second
 BRAS—Broadband Remote Access Server
 BRI—Basic Rate Interface (ISDN)
 BSP— Board Support Package
 BTU—British Thermal Units

C

CA—Certificate Authority
 CAC—Call Administration Control
 CALEA—Communications Assistance to Law Enforcement Agencies
 CAM—Content Addressable Memory
 CARP—Cache Array Routing Protocol (to replace ICP)
 CAS—Cordless Access Service
 C-BGF—Core Border Gateway Function

CBR—Constant Bit Rate (ATM)
CC—Country Code/ Call Content
CCC—Call Content Connection/ Handover Interface 3 (Call Content)
CCCid—Call Content Connection Identifier
CCM—Cisco Call Manager
CD—Call Data
CDC—Call Data Connection/ Handover Interface 2 (Call Data)
CDPN—Called Party Number
CDR—Call Detail Record
CE—Conformité Européenne (The CE marking is a European proof of conformity and is also described as "passport" that allows manufacturers and exporters to circulate products freely within the EU.)
CFM—Cubic Feet per Minute (fan speed)
CFU—Call Forwarding Unconditional
CGI—Common Gateway Interface
CIC—Carrier Indicator Code/Carrier Identification Code
CID—Caller Identification
CISSP—Certified Information Systems Security Professional
CLC—Close Logical Channel
CLF—Connectivity Session Location and Repository Function
CLCAck—Close Logical Channel Ack
CLEC—Competitive Local Exchange Carrier
CLI—Command-line Interface
Client-SI—Client-Server Information
CMS—Call Management Server
CNM—Customer Network Management
CO—Connection Oriented
CODEC—Coder/Decoder
COPS—Common Open Policy Service
CoS—Class of Service
CP—Communications Processor
CPL—Call Processing Language
CPLD—Complex Programmable Logic Device
CPM—Communications Processor Module
CPU—Central Processing Unit
CRA—Call Routing Apparatus
CRI—Call Related Information

CRLF—Carriage Return Line Feed
CS—Circuit Switch
CSA—Client Server Architecture
CSPDN—Circuit Switched Public Data Network
CSU—Channel Service Unit
CT—Cordless Telephone
CT-1—European analogue cordless telephone system
CT-2—Second-generation cordless telephone, Digital
CTI—Computer Telephony Integration
CUG—Closed User Group

D

DA—Destination Address
DAM—Data Access Method; Data Asset Management
DDD—Direct Distance Dialing
DDF—Digital Distribution Frame
DECT—Digital European Cordless Telephone
DER—Distinguished Encoding Rules
DES—Data Encryption Standard
DHCP—Dynamic Host Configuration Protocol
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
DSA—Digital Signature Algorithm
DSCP—DiffServ Codepoint
DSL—Digital Subscriber Line
DSLAM—Digital Subscriber Line Access Multiplexer

DSP—Digital Signal Processing
 DSS—Digital Satellite System
 DSU—Digital Service Unit
 dTCP—Dynamic Transmission Control Protocol
 DTD—Document Type Definition
 dTLS—Dynamic Transport Layer Security
 DTMF—Dial Tone Multi-Frequency
 DWA—Device Watchdog Answer
 DWR—Device Watchdog Request

E

ED—Ending Delimiter
 EEPROM—Electrically Erasable Programmable Read-Only Memory
 EFTPOS—Electronics Funds Transfer Point of Sale
 EGP—Exterior Gateway Protocol
 EMB—Early Media Blocking
 EMC—Electromagnetic Compatibility
 EMI—Electromagnetic Interference
 EMS—Element Management System (Acme Packet)
 ENUM—Refers to the use of an E.164 number, in reverse, with domain notation (i.e., dotted).
 EPROM—Erasable Programmable Read-Only Memory
 ER—Edge Router
 ESD—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
 FS—Fast-start
 FSA—Foreign SIP Agent (ACME-specific term?)
 FTP—File Transfer Protocol
 FTR—Flow Transform Record

G

GA—Global Address

GB—Gigabyte
GBPS—Gigabits Per Second
GigE—Gigabit Ethernet
GK—Gatekeeper
GMII—Gig Media Independent Interface
GNU—GNUs not UNIX
GOSIP—Government Open Systems Interconnection Profile
GRUU—Globally Routable User Agent URI
GPS—Global Policy Server/Global Positioning System
GSA—Global System Administrator
GSM—Global Systems for Mobile Communications
GSTN—Global Switched Telephone Network
GW—Gateway

H

HA—High Availability (Acme Packet redundancy solution)
HNT—Hosted NAT Traversal (Acme Packet)
HTML—Hypertext Markup Language
HTTP—Hypertext Transport Protocol

I

IAD—Integrated Access Device
IANA—Internet Assigned Numbers Authority
I-BCF—Interconnect Border Control Function
ICE—In Circuit Emulator
ICMP—Internet Control Message Protocol
ICP—Internet Cache Protocol
I-CSCF—Interrogating Call Session Control Function
IDS—Intrusion Detection System
IE—Information Element
IEC—International Electrotechnical Commission
IEEE—Institute of Electrical and Electronics Engineers
IESG—Internet Engineering Steering Group
IETF—Internet Engineering Task Force
IGP—Interior Gateway Protocol
IIS—Internet Information Server
IKE—Internet Key Exchange
ILEC —Independent Local Exchange carrier (USA)

IM—Instant Messaging
 IMS—IP Multimedia Subsystem
 IN—Intelligent Network
 I/O—Input/Output
 IOS—Internetworking Operating System
 IP—Internet Protocol (IPv4, IPv6)
 IPC—Inter-process Communication
 IPDR—Internet Protocol Data Record
 IPSec—Internet Protocol Security
 IPtel—Internet Protocol Telephony
 IPv—Internet Protocol version
 IS—Intercept Server
 ISDN—Integrated Services Digital Network
 ISO—International Organization of Standardization
 ISP—Internet Service Provider
 ITAD—Internet Telephony Administrative Domain
 ITSP—Internet Telephony Service Provider
 ITU—International Telecommunication Union
 ITU-T—ITU Telecommunication Standardization Sector
 IVR—Interactive Voice Response
 IWF—Interworking Function (referring to the Net-Net SBC's SIP-H.323 interworking)
 IXC—Interexchange Carrier

J

JTAG—Joint Test Action Group
 JTAPI—Java telephony application programming interface

K

Kb—Kilobits
 KB—Kilobytes
 Kbps—Kilobits per second
 KEA—Key Exchange Algorithm
 KTS—Key Telephone System

L

LA—Local Address
 LAES—Lawfully Authorized Electronic Surveillance
 LAN—Local Area Network

LATA—Local Access and Transport Area
LCD—Liquid Crystal Display
LDAP—Lightweight Direct(ory) Access Protocol
LEA—Law Enforcement Agency
LEAF—Law Enforcement Administrative Function
LEC—Local Exchange Carrier
LED—Light Emitting Diode
LEM—Local Element Manager (Acme Packet)
LEN—Local Exchange Node
LI—Lawful Intercept/ Legal Interception
LNP—Local Number Portability
LRT—Local Routing Table
LOS—Loss of Signal
LS—Location Server
LSB—Least Significant Bit
LSR—Label-switching router

M

MAC— Media Access Control/ Message Authentication Code
MAN— Metropolitan Area Networks
Mb—Megabits
MB—Megabytes
MBCD—Middlebox Control Daemon (Acme Packet)
Mbone—Multicast Backbone
MC—Monitoring Center
MCU—Multi-party Conference Unit
MD5—Message Digest 5 (hash function)
MF—Media Firewall
MG—Media Gateway
MGCP—Media Gateway Communication Protocol
MHz—Megahertz
MIB—Management Information Base
MIB II—Management Information Base II
MIBOC —Middlebox Control Protocol
MIDCOM—Middle Box Communications
MIKEY—Multimedia Internet Keying
MIME—Multipurpose Internet Mail Extension

MOC—Mandatory, Optional, Conditional
 MoIP—Messaging over Internet Protocol
 MP—Main Processor
 µP—Microprocessor (**subsystem**)
 MPLS—Multi-protocol Label Switching
 MR—Media Router
 MRCP—Media Router Control Protocol
 MSB—Most Significant Bit
 MSD—Master-Slave Determination
 MTA—Message Transfer Agent / Multimedia Terminal Adapter
 MTBF—Mean Time Between Failures
 MTTR—Mean Time To Repair
 MTU—Maximum Transmission Unit
 MX—Mail Exchange

N

N-ACD—Network Automotive Call Distribution
 NANP—North American Numbering Plan
 NAPT—Network Address Port Translation
 NAPTR—Naming Authority Pointer
 NAS—Network Access Security
 NAT—Network Address Translation
 Nco—Network Code of Practice
 NCP—Network Control Point
 NEBS—Network Equipment - Building Systems/Standards
 NE—Network Element
 NIC—Network Interface Card
 NMS—Network Management Station
 NP—Network Processor
 NPU—Network Processing Unit
 NSRG—Network Signaling Record Generator
 NTE —Networking Terminating Equipment
 NTP—Network Time Protocol
 NTU—Networking Terminating Unit
 NVRAM—Non-volatile Random Access Memory

O

OAM—Operation, Administration, and Maintenance

OC—Optical Carrier
OC-n—Optical Carrier transport
OCSP—Online Certificate Status Protocol
OCx—Optical Carrier level
OEI—Optical Electrical Interface
OEM—Original Equipment Manufacturer
OID—Object Identifier
OLC—Open Logical Channel
OLCAck—Open Logical Channel Ack
ONP—Open Network Provision
OS—Operating System
OSP—Open Settlement Protocol
OSPF—Open Shortest Path First
OSS—Operations Support Systems

P

PABX—Private Automatic Branch Exchange
PAC—Performance, Availability, Capacity (Acme Packet)
PACS—Personal Access Communications Systems
PAT—Port Address Translation
PBX—Private Branch Exchange
P-CSCF—Proxy-Call Session Control Function
PCB—Printed Circuit Board
PCDATA—Parseable Data Characters
PCI—Peripheral Component Interconnect
PCMCIA—Personal Computer Memory Card International Association
PCN—Personal Communications Network
PCS—Personal Communications Services
PD—Packet Data
PDCS—Packet Cable Distributed Call Signaling
PDH—Plesiochronous Digital Hierarchy
PDN—Public Data Network / Packet Data Network
PDP—Policy Decision Point
PDU—Protocol Data Unit (or Packet Data Unit)
PEM—Privacy Enhanced Mail
PEP—Policy Enforcement Point/Protocol Extensions Protocol
Perl—Practical Extraction Report Language

PHY—Physical Layer Device
 PIB—Policy Information Base
 PING—Packet Internet Groper
 PINT—PSTN and IP Internetworking
 PKCS-7—RFC 2315, Cryptographic Message Syntax, Version 1.5
 PKCS-10—RFC 2314, Certificate Request Syntax, Version 1.5
 PKI—Public Key Infrastructure
 PMC—PCI Mezzanine Card
 PNNI—Private Network Node Interface (ATM)
 PNO—Public Network Operator
 POP—Point of Presence
 POS—Packet Over SONET
 POTS—Plain Old Telephone Service
 PPP—Point-to-Point Protocol
 PROM—Programmable Read-Only Memory
 PS—Policy Server
 PSAP—Public Safety Answering Point
 PSTN—Public Switched Telephone Network (Telecom Network)
 PTE—Packet Transform Engine
 PTO—Public Telecommunications Operator
 PTT—Post, Telephone, and Telegraph
 PWB—Printed Wiring Board

Q

QoS—Quality of Service
 QSIG—Unified International Digital Corporate Network Signaling Standard

R

RACF—Resource and Admission Control Function
 RADIUS—Remote Authentication Dial-in User Service
 RAM—Random Access Memory
 RARP—Reverse Address Resolution Protocol
 RAS—Remote Access Service; Registration Admission and Status (H.323)
 RC—Registration Cache
 RC2 and RC4—Rivest encryption ciphers developed for RSA Data Security
 RED—Random Early Discard
 REN—Ringer Equivalent Number
 RFC—Request for Comments

RIP—Routing Information Protocol
 RISC—Reduced Instruction Set Chip
 RMON—Remote (Network) Monitoring
 ROM—Read-Only Memory
 RPC—Remote Procedure Call
 RR—Received Report
 RS-232—Recommended Standard 232 (computer serial interface, IEEE)
 RSA—Rivest, Shamir, & Adleman (public key encryption technology)
 RSIP—ReStart In Progress
 RSVP—Resource Reservation Protocol
 RTCP—Real-time Control Protocol
 RTP—Real-time Transport Protocol
 RTP/AVP—Real-time Transport Protocol/Audio-Video Protocol
 RTSP—Real-time Streaming Protocol
 RTT—Round Trip Time

S

SA—Source Address / Session Agent / Security Association
 SAG—Session Agent Group
 SBC—Session Border Controller
 SCE—Service Control Environment
 SCP—Service Control Point
 S-CSCF—Serving Call Session Control Function
 SCTP—Streaming Control Transmission Protocol
 SDES—Source Description RTCP (Real-Time Control Protocol) Packet
 SDH—Synchronous Digital Hierarchy
 SDP—Session Description Protocol
 SDRAM—Synchronous Dynamic Random Access Memory
 SERDES—Serial De-serializer
 SFE—Security Front End
 SHA-1—Secure Hash Algorithm, a hash function used by the U.S. Government
 SIG—Special Interest Group
 SIM—Subscriber Identity Module
 SIMM—Single In-line Memory Module
 SIP—Session Initiation Protocol
 SLA—Service Level Agreement
 SME—Small to Medium Enterprise(s)

SMIL—Synchronized Multimedia Integration Language
 SMP—Simple Management Protocol
 SMTP—Simple Mail Transfer Protocol
 SNMP—Simple Network Management Protocol
 SOCKS—SOCKetS server
 SONET—Synchronous Optical Network
 SP—Source Port / Service Provider
 SR —(Net-Net) Session Router
 SRAM—Static Random Access Memory
 SRS—Session Routing System
 SRTP—Secure Real-Time Transport Protocol
 SRV—Resource record for servers (DNS)
 SS—Slow-start
 SS7—Signaling System 7
 SSH—Secured Shell or Secure Socket Shell
 SSL—Secure Socket Layer
 SSP—Service Switching Point
 sTCP—Static Transmission Control Protocol
 STL—Standard Template Library
 sTLS—Static Transport Layer Security
 STP—Signal Transfer Point; Service Transfer Point
 SVC—Signaling Virtual Channel (ATM) / Switched Virtual Circuit (Packet Switching)

T

TA—Terminal Adapter (ISDN)
 TAC—Terminal Access Control
 TACACS+—Terminal Access Controller Access Control System
 TAPI—Telephony Application Program Interface
 TAXI—Transparent Asynchronous Transmitter/Receiver Interface
 TCB—Task Control Bar/Task Control Block
 TCI—Tag Control Identifier
 TCP—Transmission Control Protocol
 TCP/IP—Transmission Control Protocol /Internet Protocol
 TCS—Terminal Capability Set
 TEN—Transit Exchange Node
 TFTP—Trivial File Transfer Protocol

TLS—Transport Layer Security (same as SSL)
TLV—Tag Length Value
TM—Traffic Manager
TMN—Telecommunications Management Network
ToS—Type of Service
TRIB—Telephony Routing Information Base
TRIP—Telephony Routing over IP
TS—Time Slot
TSAP—Transport Service Access Point
TSAPI—Telephony Server API
TTL—Time to Live
TTR—Time to Resume

U

UA—User Agent
UAC—User Agent Client
UAS—User Agent Server
UDP—User Datagram Protocol
UE—User Equipment
UL—Underwriters Laboratories
UMTS—Universal Mobile Telecommunications Systems
UNI—User-to-Network Interface
UPS—Uninterruptible Power Supply
UPT—Universal Portable Telephone
URI—User Resource Identifier
URL—Uniform Resource Locator
UTC—Coordinated Universal Time
UTP—Unshielded Twisted Pair

V

VAC—Volts Alternating Current
VANS—Value Added Network Services
VAR—Value Added Reseller
VarBind—Variable Binding
VBR—Variable Bit Rate
VC—Virtual Channel (ATM)/Virtual Container (SDH)
VCC—Virtual Channel Connection (ATM)
VCI—Virtual Channel Identifier

VDC—Volts Direct Current
 VFD—Vacuum Florescent Display
 VLAN—Virtual Local Area Network
 VLL—Virtual Leased Lines
 VoIP—Voice Over Internet Protocol
 VP—Virtual Path
 VPC—Virtual Path Connection
 VPI—Virtual Path Identifier
 VPN—Virtual Private Network
 VSA—Vendor-specific Attribute (RADIUS extension)
 VTOA—Voice and Telephony over ATM

W

WAN—Wide Area Network
 WLL—Wavelength Division Multiplex

X

XE—Translation Engine
 XML—Extensible Markup Language
 XSM—External Search Machine

Y

(None to list.)

Z

((None to list.)

Signaling Protocol Acronyms

The acronyms used in this guide's discussion of H.323 signaling services and IWF services.

H.323

We use the acronyms listed below to refer to H.323 signaling messages and other related H.323 behavior.

ACF—Admission Confirm
 Alerting—Message used when called party alerted
 ARQ—Admission Request
 Call Proceeding—Message used when call established
 CLC—Close Logical Channel
 CLC Ack—Close Logical Channel Acknowledgment
 Connect—Message used when called party accepts call

CPN—Calling party Number
 DRQ—Delete Request State
 GRQ—Gatekeeper Discovery
 IRQ—Information Request
 IRR—Information Request Response
 LCF—Location Confirm
 LRJ—Location Reject
 LRQ—Location Request
 MSD—Master/Slave Determination
 OLC—Open Logical Channel
 OLC Ack—Open Logical Channel Acknowledgment
 RAS—Registration, Admission, and Status
 RCF—Registration Confirm
 Release Complete—Message used when call is released, signaling channel open
 RRJ—Registration Reject
 RRQ—Registration Request
 Setup—Message used to request connection
 TCS—Terminal Capability Set
 UCF—Unregistration Confirm
 URJ—Unregistration Reject
 URQ—Unregistration Request
 VGW—Virtual Gateway

MGCP

AUCX—Audit Connection
 AUEP—Audit Endpoint
 CRCX—Create Connection
 DLCX—Delete Connection
 MDCX—Modify Connection
 NCS—Network Call Signaling
 NTFY—Notify
 PVT—Program Value Tree
 RQNT—Request for Notification
 RSIP—Restart In Progress

SIP

ACK—Acknowledgement (SIP)
 DMR—Distributed Media Release

MGW—Media Gateway

OSI—Open System Interconnect(ion)

TGRP—Trunk Group Name

TISPAN—Telecom and Internet converged Services and Protocols for Advanced Networks

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 Configuration Elements
Access Control	bootparams
Accounting Config	
Authentication	
Certificate Record	
Class Profile	
Codec Policy	
DNS ALG Service-	
DNS Config	
Enum	
External Policy Server	
H.323	<p>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. For new calls, the changes will be in effect.</p> <ul style="list-style-type: none">• 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 Configuration Elements
IPSEC	
IWF	
Licensing	
Local Policy	
Local Response Map	
Local Routing Config	
Media Manager	The Media Manager element is supported with the exception of the following parameters: <ul style="list-style-type: none">• red-flow-port• red-max-trans• red-sync-start-time• red-sync-comp-time• red-mgcp-port
Media Policy	
Media Profile	
MGCP	
Network Interface	
Net Management Control	
Network Parameters	
NTP Sync	
Packet Trace Config	
Physical Interface	
Q850 SIP Map	
Realm Config	
Redundancy Config	The Redundancy Config element is supported with the exception of the following parameters: <ul style="list-style-type: none">• state• port• cfg-port• cfg-max-trans• cfg-sync-start-time• cfg-sync-comp-time
Session Agent	
Session Group	
Session Router	
Session Translation	
Session Constraints	

ACLI Supported Configuration Elements	ACLI Unsupported Configuration Elements
SIP Config	The SIP Config element is supported with the exception of the following parameters: <ul style="list-style-type: none">• red-sip-port• red-max-trans• red-sync-start-time• red-sync-stop-time
SIP Feature	
SIP Interface	collect>boot-state
SIP Manipulation	
SIP NAT	
SIP Q850 Map	
SIP Response Map	
SNMP	
Static Flow	
Steering Pool	
Surrogate Agent	
System	The System Config element is supported with the exception of the following parameters: <ul style="list-style-type: none">• options
Test Pattern Rule	
Test Policy	
Test Translation	
TLS Global	
TLS Profile	
Translation Rules	
Trap Receiver	

