



Cisco ASR 5500 System Administration Guide

Version 14.0

Last Updated May 31, 2013

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-27230-04

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco ASR 5500 System Administration Guide

© 2013 Cisco Systems, Inc. All rights reserved.

CONTENTS

About this Guide	xiii
Conventions Used	xiv
Supported Documents and Resources	xv
Related Documentation.....	xv
Contacting Customer Support.....	xvi
System Operation and Configuration	17
Terminology.....	18
Contexts	18
Ports	18
Logical Interface.....	18
Management Interface	19
Bindings.....	19
Services.....	19
AAA Servers.....	20
Subscribers	20
How the System Selects Contexts	21
Context Selection for Context-level Administrative User Sessions.....	21
Context Selection for Subscriber Sessions.....	23
Understanding the ASR 5500 Boot Process	24
Understanding Configuration Files	26
IP Address Notation.....	28
IPv4 Dotted-Decimal Notation.....	28
IPv6 Colon-Separated-Hexadecimal Notation	28
CIDR Notation	28
Alphanumeric Strings	30
Character Set	30
Quoted Strings	31
Getting Started	33
ASR 5500 Configuration.....	34
Using the ASR 5500 Quick Setup Wizard	34
Using the CLI for Initial Configuration.....	40
Configuring the System for Remote Access.....	43
Configuring the Management Interface with a Second IP Address.....	45
System Settings	47
Configuring a Second Management Interface.....	48
Verifying and Saving Your Interface and Port Configuration.....	49
Configuring System Timing	50
Setting the System Clock and Time Zone.....	50
Verifying and Saving Your Clock and Time Zone Configuration.....	50
Configuring Network Time Protocol Support.....	51
Configuring NTP Servers with Local Sources.....	52
Using a Load Balancer	52
Verifying the NTP Configuration.....	52
Enabling CLI Timestamping	54
Configuring System Administrative Users	54

Configuring Context-level Administrative Users	54
Configuring Context-level Security Administrators.....	54
Configuring Context-level Administrators.....	55
Configuring Context-level Operators.....	55
Configuring Context-level Inspectors	56
Verifying Context-level Administrative User Configuration	56
Configuring Local-User Administrative Users.....	57
Verifying Local-User Configuration.....	57
Configuring TACACS+ for System Administrative Users	58
Operation	58
User Account Requirements	58
TACACS+ User Account Requirements	58
User Account Requirements	59
Configuring TACACS+ AAA Services	60
Verifying the TACACS+ Configuration	61
Configuring a Chassis Key	62
Configuring MIO/UMIO Port Redundancy	63
Configuring MIO/UMIO Port Redundancy Auto-Recovery	65
Verifying Port Redundancy Auto-Recovery	66
Configuring Data Processing Card (DPC) Availability	67
Verifying Card Configurations	67
Configuring ASR 5500 Link Aggregation.....	68
LAG and Master Port.....	68
LAG and Port Redundancy	68
LAG and Multiple Switches	68
Multiple Switches with L2 Redundancy.....	69
Port States for Auto-Switch	69
Hold Time.....	70
Preferred Slot	70
Auto-Switch Criteria	70
Link Aggregation Control	70
Redundancy Options.....	72
Horizontal Link Aggregation with Two Ethernet Switches.....	72
Link Aggregation Status	72
Management Settings	75
ORBEM and the Web Element Manager.....	76
Configuring ORBEM Client and Port Parameters	76
Configuring IIOIP Transport Parameters.....	77
Verifying ORBEM Parameters.....	78
SNMP Support.....	79
Configuring SNMP and Alarm Server Parameters.....	79
Verifying SNMP Parameters	80
Controlling SNMP Trap Generation.....	81
Verifying and Saving Your Configuration	83
Verifying the Configuration	84
Feature Configuration.....	84
Service Configuration	84
Context Configuration.....	85
System Configuration	85
Finding Configuration Errors	85
Saving the Configuration	86
System Interfaces and Ports.....	89

Creating Contexts	90
Viewing and Verifying Contexts	90
Creating and Configuring Ethernet Interfaces and Ports	91
Creating an Interface	91
Configuring a Port and Binding It to an Interface	92
Configuring a Static Route for an Interface	92
Viewing and Verifying Port Configuration	93
System Security	95
Per-Chassis Key Identifier	96
MIO/UMIO Synchronization	96
Protection of Passwords	96
Secure Configuration Password Encryption	97
Support for Non-Current Encryptions and Decryptions	97
Support for ICSR Configurations	97
Encrypted SNMP Community Strings	98
Lawful Intercept Restrictions	98
LI Server Addresses	98
Modifying Intercepts	98
Adding, Modifying and Removing Users	99
Notification of Users Being Added or Deleted	99
Notification of Changes in Privilege Levels	99
User Access to Operating System Shell	99
Hidden Commands	100
Enabling cli test-commands Mode	100
Enabling Password for Access to CLI-test commands	100
Exec Mode cli test-commands	101
Configuration mode cli test-commands	101
Software Management Operations	103
Understanding the Local File System	104
File Types Used by the Local File System	104
Understanding the boot.sys File	104
Maintaining the Local File System	106
File System Management Commands	106
Synchronizing the File System	106
Creating Directories	107
Renaming Files and Directories	107
Copying Files on the ASR 5500 Chassis	107
Deleting Files	109
Removing Directories	109
Formatting Local Devices	110
Applying Pre-existing CLI Configuration Files	110
Viewing Files on the Local File System	111
Viewing the Contents of a Local Device	111
Viewing CLI Configuration and boot.sys Files	111
Validating an Operating System File	112
Configuring the Boot Stack	113
System Boot Methods	113
Viewing the Current Boot Stack	113
Adding a New Boot Stack Entry	114
Deleting a Boot Stack Entry	115
Network Booting Configuration Requirements	116
Configuring the Boot Interface	116
Configuring the Boot Network	117

Configuring Boot Network Delay Time	118
Configuring a Boot Nameserver	118
Upgrading the Operating System Software	119
Identifying OS Release Version and Build Number	119
Verify Free Space on the /flash Device	119
Download the Software Image from the Support Site	120
Transfer StarOS Image to /flash on the Chassis	120
Saving a Copy of the Current Configuration File	121
Off-line Software Upgrade	121
Configure a Newcall Policy	121
Configure a	122
Back up the Current CLI Configuration File	122
Create a New Boot Stack Entry	123
Synchronize File Systems	123
Reboot the Chassis	123
Verify the Running Software Version	124
Restoring the Previous Software Image	124
Upgrading ICSR Chassis	124
Performing Dynamic Software Updates	125
Overview	125
Downloading the Patch Kit	125
Unpacking the Patch Kit	126
Configuring the Plugin	126
Loading the Plugin	127
Rolling Back to a Previous Plugin Version	127
DSU show Commands	127
Managing License Keys	129
New System License Keys	129
Session Use and Feature Use Licenses	129
Installing New License Keys	129
Cutting and Pasting the Key	129
Adding License Keys to Configuration Files	130
License Expiration Behavior	131
Requesting License Keys	131
Viewing License Information	131
Deleting a License Key	132
Management Card Replacement and License Keys	132
Managing Local-User Administrative Accounts	133
Configuring Local-User Password Properties	133
Configuring Local-User Account Management Properties	133
Local-User Account Lockouts	133
Local-User Account Suspensions	134
Changing Local-User Passwords	134
Monitoring the System	135
SNMP Notifications	136
Monitoring System Status and Performance	136
Clearing Statistics and Counters	137
Monitoring ASR 5500 Hardware Status	138
Bulk Statistics	141
Configuring Communication with the Collection Server	142
Configuring Standard Settings	142
Configuring Optional Settings	142
Configuring Bulk Statistic Schemas	143

Verifying Your Configuration	143
Saving Your Configuration	144
Viewing Collected Bulk Statistics Data	145
Manually Gathering and Transferring Bulk Statistics	146
Clearing Bulk Statistics Counters and Information	146
Bulk Statistics Event Log Messages	146
System Logs	147
System Log Types	148
Configuring Event Logging Parameters	149
Configuring Event Log Filters	149
Configuring syslog Servers	150
Configuring Active Logs	151
Specifying Facilities	152
Configuring Trace Logging	159
Configuring Monitor Logs	160
Enabling Monitor Logs	160
Disabling Monitor Logs	160
Viewing Logging Configuration and Statistics	161
Viewing Event Logs Using the CLI	162
Configuring and Viewing Crash Logs	163
Configuring Software Crash Log Destinations	163
Viewing Abridged Crash Log Information Using the CLI	164
Saving Log Files	166
Event ID Overview	167
Event Severities	175
Understanding Event ID Information in Logged Output	175
Troubleshooting	177
Detecting Faulty Hardware	178
Licensing Issues	178
Using the CLI to View Status LEDs	178
Checking the LEDs on the PFU	179
Checking the LEDs on the MIO and UMIO	180
MIO/UMIO Run/Fail LED States	180
MIO/UMIO Active LED States	181
MIO/UMIO Redundancy LED States	182
MIO/UMIO Master LED States	182
MIO/UMIO Busy LED States	183
MIO/UMIO – Interface Link LED States	183
MIO/UMIO – Interface Activity LED States	183
Checking the LEDs on the DPC and UDPC	184
DPC/UDPC Run/Fail LED States	184
DPC/UDPC Active LED States	185
DPC/UDPC Redundancy LED States	186
Checking the LEDs on the FSC	187
FSC Run/Fail LED States	187
FSC Active LED States	188
FSC Redundancy LED States	189
FSC Drive n Activity LED States	189
Checking the LEDs on the SSC	190
SSC Run/Fail LED States	190
SSC Active LED States	191
SSC Redundancy LED States	192
SSC System Status LED States	192

SSC System Service LED States	193
Testing System Alarm Outputs.....	193
Taking Corrective Action.....	194
Switching MIO/UMIOs	194
Busying Out a DPC/UDPC	194
Migrating a DPC/UDPC	195
Halting Cards.....	196
Initiate a Card Halt	196
Restore a Previously Halted Card.....	196
Verifying Network Connectivity	198
Using the ping or ping6 Command.....	198
Syntax	198
Troubleshooting	199
Using the traceroute or traceroute6 Command.....	200
traceroute – IPv4.....	200
traceroute6 – IPv6.....	201
Viewing IP Routes	202
Viewing the Address Resolution Protocol Table	203
Using the System Diagnostic Utilities	204
Using the Monitor Utility.....	204
Using the Protocol Monitor	204
Using the Protocol Monitor for a Specific Subscriber	205
Using the DHCP Testing Tool	207
System Recovery	209
Prerequisites.....	210
Console Access.....	210
Boot Image	210
Accessing the boot CLI.....	211
Initiate a Reboot	211
Interrupt the Boot Sequence	211
Enter CLI Mode	211
boot Command Syntax.....	212
Bootting from a Selected Image	213
Boot Using No Configuration File.....	213
Boot Using A Specified Configuration File	213
Engineering Rules.....	215
CLI Session Rules	216
ASR 5500 Interface and Port Rules	216
Packet Data Network (PDN) Interface Rules	216
Context Rules	217
Subscriber Rules	219
Service Rules.....	219
Access Control List (ACL) Engineering Rules	220
System Software Tasks.....	221
Overview.....	222
Primary Task Subsystems.....	223
Primary Subsystem Controllers and Managers	224
ASR 5x00 Subsystems	225
Access Control Lists	245
Overview.....	246
Understanding ACLs.....	246
Rule(s)	246

Actions	246
Criteria.....	247
Rule Order.....	248
Configuring ACLs on the System	249
Creating ACLs	249
Configuring Action and Criteria for Subscriber Traffic.....	250
Configuring an Undefined ACL	250
Verifying the ACL Configuration	251
Applying IP ACLs.....	251
Applying an ACL to an Individual Interface	253
Applying the ACL to an Interface	253
Verifying the ACL Configuration on an Interface.....	254
Applying an ACL to All Traffic Within a Context.....	254
Applying the ACL to a Context	255
Verifying the ACL Configuration in a Context	255
Applying an ACL to a RADIUS-based Subscriber	256
Applying an ACL to an Individual Subscriber	256
Verifying the ACL Configuration to an Individual Subscriber	257
Applying a Single ACL to Multiple Subscribers.....	258
Applying an ACL to the Subscriber Named default	258
Applying an ACL to Service-specified Default Subscriber.....	260
Congestion Control.....	265
Overview.....	266
Configuring Congestion Control	267
Configuring the Congestion Control Threshold.....	267
Configuring Service Congestion Policies	268
Configuring Overload Reporting on the MME (ASR 5000 only)	268
Enabling Congestion Control Redirect Overload Policy.....	269
Verify the Service Overload Policies.....	269
Verify the Congestion Control Configuration	269
Verify MME Congestion Action Profiles	271
Disconnecting Subscribers Based on Call or Inactivity Time	271
Content Service Steering	273
Overview.....	274
Configuring Internal Content Service Steering	274
Defining IP Access Lists for Internal CSS	275
Applying an ACL to an Individual Subscriber (Optional)	275
Applying an ACL to Multiple Subscribers (Optional)	275
Applying an ACL to the Subscriber Named default (Optional)	276
Applying an ACL to Service-specified Default Subscribers (Optional)	276
Applying an ACL to Multiple Subscribers via APNs (Optional)	276
Interchassis Session Recovery	277
Overview.....	278
Interchassis Communication	278
Checkpoint Messages	278
AAA Monitor	278
BGP Interaction	279
Requirements	279
ICSR Operation	280
Chassis Initialization.....	283
Chassis Operation.....	283
Chassis Communication	283

Chassis Switchover.....	283
Configuring Interchassis Session Recovery (ICSR).....	284
Configuring the Service Redundancy Protocol (SRP) Context.....	285
Creating and Binding the SRP Context.....	285
Configuring the SRP Context Parameters.....	285
Configuring the SRP Context Interface Parameters.....	286
Verifying SRP Configuration.....	287
Modifying the Source Context for ICSR.....	288
Configuring BGP Router and Gateway Address.....	288
Configuring the SRP Context for BGP.....	288
Verifying BGP Configuration.....	289
Modifying the Destination Context for ICSR.....	289
Configuring BGP Router and Gateway Address in Destination Context.....	289
Configuring SRP Context for BGP for Destination Context.....	289
Setting Subscriber to Default Mode.....	290
Verifying BGP Configuration in Destination Context.....	290
Disabling Bulk Statistics Collection on a Standby System.....	290
Verifying the Primary and Backup Chassis Configuration.....	290
Updating the Operating System.....	291
Both ICSR Chassis.....	296
Downloading and Transferring the StarOS Build.....	296
Standby Backup Chassis.....	297
Performing Health Checks.....	297
Performing SRP Checks.....	297
Performing BGP Checks.....	297
Updating the Boot Record.....	297
Synchronizing File Systems.....	298
Reloading the Chassis.....	298
Updating the Configuration File.....	298
Verifying the Software Version.....	298
Saving the Configuration File.....	298
Completing the Update Process.....	299
Waiting for Session Synchronization.....	299
Primary Chassis.....	300
Initiating an SRP Switchover.....	300
Checking AAA Monitor Status on the Newly Active Chassis.....	300
Completing the Software Update.....	300
Initiating an SRP Switchover.....	301
Checking AAA Monitor Status.....	301
Making Test Calls.....	301
Fallback Procedure.....	301
Routing.....	303
Routing Policies.....	304
Creating IP Prefix Lists.....	304
Creating Route Access Lists.....	304
Creating AS Path Access Lists.....	305
Creating Route Maps.....	305
Sample Configuration.....	306
Static Routing.....	307
Adding Static Routes to a Context.....	307
Deleting Static Routes From a Context.....	308
OSPF Routing.....	309
OSPF Version 2 Overview.....	309
Basic OSPFv2 Configuration.....	310

Enabling OSPF Routing For a Specific Context	310
Enabling OSPF Over a Specific Interface.....	310
Redistributing Routes Into OSPF (Optional).....	310
Confirming OSPF Configuration Parameters.....	311
OSPFv3 Routing.....	312
OSPFv3 Overview.....	312
Basic OSPFv3 Configuration	312
Enabling OSPFv3 Routing For a Specific Context	312
Enabling OSPFv6 Over a Specific Interface.....	312
Redistributing Routes Into OSPFv3 (Optional)	313
Confirming OSPFv3 Configuration Parameters	313
Viewing Routing Information.....	314
Equal Cost Multiple Path (ECMP)	314
BGP-4 Routing	315
Overview of BGP Support	315
Configuring BGP	316
Redistributing Routes Into BGP (Optional)	316
Bidirectional Forwarding Detection.....	317
Overview of BFD Support	317
Configuring BFD.....	317
Configuring a BFD Context.....	318
Configuring IPv4 BFD for Static Routes	318
Configure IPv6 BFD for Static Routes	319
Configuring BFD for Single Hop	319
Configuring Multihop BFD.....	320
Scaling of BFD.....	320
Associate BGP Neighbors with the Context	321
Associate OSPF Neighbors with the Context	321
Associate BFD Neighbor Groups with the BFD Protocol.....	321
Session Recovery	323
How Session Recovery Works	324
Additional Hardware Requirements.....	326
Configuring the System to Support Session Recovery	327
Enabling Session Recovery	327
Enabling Session Recovery on an Out-of-Service System	327
Enabling Session Recovery on an In-Service System	328
Disabling the Session Recovery Feature	329
Viewing Session Recovery Status	329
Viewing Recovered Session Information	330
VLANs.....	333
Overview.....	334
Overlapping IP Address Pool Support – GGSN.....	334
RADIUS VLAN Support – Enhanced Charging Services.....	334
APN Support – PDN Gateway (P-GW)	335
Creating VLAN Tags.....	335
Verifying the Port Configuration	336
Configuring Subscriber VLAN Associations	337
RADIUS Attributes Used	337
Configuring Local Subscriber Profiles	337
Verify the Subscriber Profile Configuration	337
VLAN-Related CLI Commands.....	338
BGP MPLS VPNs	341

Introduction	342
MPLS-CE Connected to PE.....	342
ASR 5x00 as a PE	343
Overview.....	343
Sample Configuration	343
VPN-Related CLI Commands.....	346

About this Guide

This preface describes the *System Administration Guide*, how it is organized and its document conventions.

The *System Administration Guide* describes how to generally configure and maintain StarOS running on an ASR 5500 platform. It also includes information on monitoring system performance and troubleshooting.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a <i>screen display</i>	This typeface represents displays that appear on your terminal screen, for example: <i>Login:</i>
Text represented as commands	This typeface represents commands that you enter, for example: show ip access-list This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a command variable	This typeface represents a variable that is part of a command, for example: show card slot_number slot_number is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the File menu, then click New

Command Syntax Conventions	Description
{ keyword or <i>variable</i> }	Required keywords and variables are surrounded by grouped braces. Required keywords and variables are those components that are required to be entered as part of the command syntax.
[keyword or <i>variable</i>]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by brackets.
	Some commands support alternative variables. These options are documented within braces or brackets by separating each variable with a vertical bar. These variables can be used in conjunction with required or optional keywords or variables. For example: <pre>{ nonce timestamp }</pre> OR <pre>[count <i>number_of_packets</i> size <i>number_of_bytes</i>]</pre>

Supported Documents and Resources

Related Documentation

The most up-to-date information for this product is available in the product *Release Notes* provided with each software release.

The following user documents are available on www.cisco.com:

- *ASR 5500 Installation Guide*
- *Command Line Interface Reference*
- *AAA Interface Administration and Reference*
- *GTPP Interface Administration and Reference*
- *Release Change Reference*
- *Statistics and Counters Reference*
- *SNMP MIB Reference*
- *Thresholding Configuration Guide*
- *Cisco Web Element Manager Installation and Administration Guide*
- Product-specific and feature-specific Administration guides

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of <http://www.cisco.com> for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.

Chapter 1

System Operation and Configuration

The ASR 5500 is designed to provide subscriber management services for high-capacity 4G wireless networks.

Before you connect to the command line interface (CLI) and begin system configuration, you must understand how the system supports these services. This segment provides terminology and background information to consider before you configure the system. The following sections are included:

- [Terminology](#)
- [How the System Selects Contexts](#)
- [Understanding the ASR 5500 Boot Process](#)
- [IP Address Notation](#)
- [Alphanumeric Strings](#)

Terminology

This section defines important terms used throughout this guide.

Contexts

A context is a logical grouping or mapping of configuration parameters that pertain to various physical ports, logical IP interfaces, and services. A context can be thought of as a virtual private network (VPN).

The system supports the configuration of multiple contexts. Each context is configured and operates independently of the others. Once a context has been created, administrative users can configure services, logical IP interfaces, and subscribers for that context and then bind the logical interfaces to physical ports.

You can also assign a domain alias to a context; if a subscriber's domain name matches one of the configured alias names for a context, that context is used.

Ports

Ports are the physical connectors on line cards that support remote access and subscriber traffic. Port configuration includes traffic profiles, data encapsulation methods, media type, and other information for physical connectivity between the system and the rest of the network.

Ports are identified by the chassis slot number for the Management Input/Output (MIO) or Management I/O Universal Card (UMIO) card, followed by the physical connector number. For example, Port 5/10 identifies connector number 10 on the MIO/UMIO card in slot 5.

Associate ports with contexts through bindings. For additional information on bindings, refer to the *Bindings* section below. You can configure each physical port to support multiple logical IP interfaces, each with up to 17 IP addresses (one primary and up to 16 secondaries).

For complete information on line cards and port assignments, refer to the *ASR 5500 Installation Guide*.

 **Important:** UMIO cards and UDPCs are direct replacements for MIO cards and DPCs. However, a special Universal PID license must be purchased and installed on the chassis for each installed UMIO and UDPC. Contact your Cisco account representative for additional licensing information.

 **Important:** Throughout this guide, any reference to an MIO card or DPC is assumed to also refer to the UMIO and UDPC respectively.

Logical Interface

You must associate a port with a virtual circuit or tunnel called a *logical interface* before the port can allow the flow of user data. A logical interface within the system is the assignment of a virtual router instance that provides higher-layer protocol transport, such as Layer 3 IP addressing. Interfaces are configured as part of the VPN context and are independent from the physical port that will be used to bridge the virtual interfaces to the network.

There are several types of logical interfaces to configure to support Simple and Mobile IP data applications.

Management Interface

This interface provides the point of attachment to the management network. The interface supports remote access to the command line interface (CLI). It also supports Common Object Request Broker Architecture (CORBA)-based management via the Web Element Manager application, and event notification via the Simple Network Management Protocol (SNMP).

Define management interfaces in the *local* context and bind them to the ports on the Management Input/Output (MIO) cards.

Bindings

A binding is an association between elements within the system. There are two types of bindings: static and dynamic.

Static binding is accomplished through system configuration. Static bindings associate:

- A specific logical interface (configured within a particular context) to a physical port. Once the interface is bound, traffic can flow through the context as if it were any physically-defined circuit. Static bindings support any encapsulation method over any interface and port type.
- A service to an IP address assigned to a logical interface within the same context. This allows the interface to take on the characteristics (that is, support the protocols) required by the service.

Dynamic binding associates a subscriber to a specific egress context based on the configuration of their profile or system parameters. This provides a higher degree of deployment flexibility, as it allows a wireless carrier to support multiple services and facilitates seamless connections to multiple networks.

Management ports can only be bound in the local context. Traffic or subscriber ports can only be bound in a non-local context.

Services

Configure services within a context to enable certain functionality. The following are examples of services you can configure on the system, subject to licensing availability and platform type:

- Gateway GPRS Support Node (GGSN) services
- Serving GPRS Support Node (SGSN) Services
- Packet Data Serving Node (PDSN) services
- Home Agent (HA) services
- Layer 2 Tunneling Protocol Access Concentrator (LAC) services
- Dynamic Host Control Protocol (DHCP) services
- PDN Gateway (P-GW) Services
- Serving Gateway (S-GW) Services
- Intelligent Policy Control Function (IPCF) Services (PCC-Service, PCC-Policy, PCC-AF)

AAA Servers

Authentication, Authorization and Accounting (AAA) servers store profiles, perform authentication, and maintain accounting records for each mobile data subscriber. The AAA servers communicate with the system over an AAA interface. The system supports the configuration of up to 128 interfaces to AAA servers.

It is important to note that for Mobile IP, there can be Foreign AAA (FAAA) and Home AAA (HAAA) servers. FAAA servers typically reside in the carrier's network. HAAA servers could be owned and controlled by either the carrier or the home network. If the HAAA server is owned and controlled by the home network, accounting data is transferred to the carrier via an AAA proxy server.

 **Important:** Mobile IP support depends on the availability and purchase of a standalone license or a license bundle that includes Home Agent (HA).

Subscribers

Subscribers are the end-users of the service; they gain access to the Internet, their home network, or a public network through the system.

There are three primary types of subscribers:

- **RADIUS-based Subscribers:** The most common type of subscriber, these users are identified by their International Mobile Subscriber Identity (IMSI) number, an Electronic Serial Number (ESN), or by their domain name or user name. They are configured on and authenticated by a RADIUS AAA server.

Upon successful authentication, various attributes that are contained in the subscriber profile are returned. The attributes dictate such things as session parameter settings (for example, protocol settings and IP address assignment method), and what privileges the subscriber has.

 **Important:** Attribute settings received by the system from a RADIUS AAA server take precedence over local-subscriber attributes and parameters configured on the system.

- **Local Subscribers:** These are subscribers, primarily used for testing purposes, that are configured and authenticated within a specific context. Unlike RADIUS-based subscribers, the local subscriber's user profile (containing attributes like those used by RADIUS-based subscribers) is configured within the context where they are created.

When local subscriber profiles are first created, attributes for that subscriber are set to the system's default settings. The same default settings are applied to all subscriber profiles, including the subscriber named *default* which is created automatically by the system for each system context. When configuring local profile attributes, the changes are made on a subscriber-by-subscriber basis.

 **Important:** Attributes configured for local subscribers take precedence over context-level parameters. However, they *could* be over-ridden by attributes returned from a RADIUS AAA server.

- **Management Subscribers:** A management user is an authorized user who can monitor, control, and configure the system through the CLI or Web Element Manager application. Management is performed either locally, through the system Console port, or remotely through the use of the Telnet or secure shell (SSH) protocols. Management users are typically configured as a local subscriber within the Local context, which is used exclusively for system management and administration. As with a local subscriber, a management subscriber's

user profile is configured within the context where the subscriber was created (in this case, the Local context). However, management subscribers may also be authenticated remotely via RADIUS, if an AAA configuration exists within the local context, or TACACS+.

How the System Selects Contexts

This section describes the process that determines which context to use for context-level administrative users or subscriber sessions. Understanding this process allows you to better plan your configuration in terms of how many contexts and interfaces you need to configure.

Context Selection for Context-level Administrative User Sessions

The system comes configured with a context called *local* that you use specifically for management purposes. The context selection process for context-level administrative users (those configured within a context) is simplified because the management ports on the MIO are associated only with the Local context. Therefore, the source and destination contexts for a context-level administrative user responsible for managing the entire system should always be the local context.

A context-level administrative user can also connect through other interfaces on the system and still have full system management privileges.

A context-level administrative user can be created in a non-local context. These management accounts have privileges only in the context in which they are created. This type of management account can connect directly to a port in the context in which they belong, if local connectivity is enabled (SSHD, for example) in that context.

For all FTP or SFTP connections, you must connect through an MIO management interface. If you SFTP or FTP as a non-local context account, you must use the username syntax of *username@contextname*.

The context selection process becomes more involved if you are configuring the system to provide local authentication or work with a AAA server to authenticate the context-level administrative user.

The system gives you the flexibility to configure context-level administrative users locally (meaning that their profile will be configured and stored in its own memory), or remotely on an AAA server. If a locally-configured user attempts to log onto the system, the system performs the authentication. If you have configured the user profile on an AAA server, the system must determine how to contact the AAA server to perform authentication. It does this by determining the AAA context for the session.

The following table and flowchart describe the process that the system uses to select an AAA context for a context-level administrative user. Items in the table correspond to the circled numbers in the flowchart.

Figure 1. Context-level Administrative User AAA Context

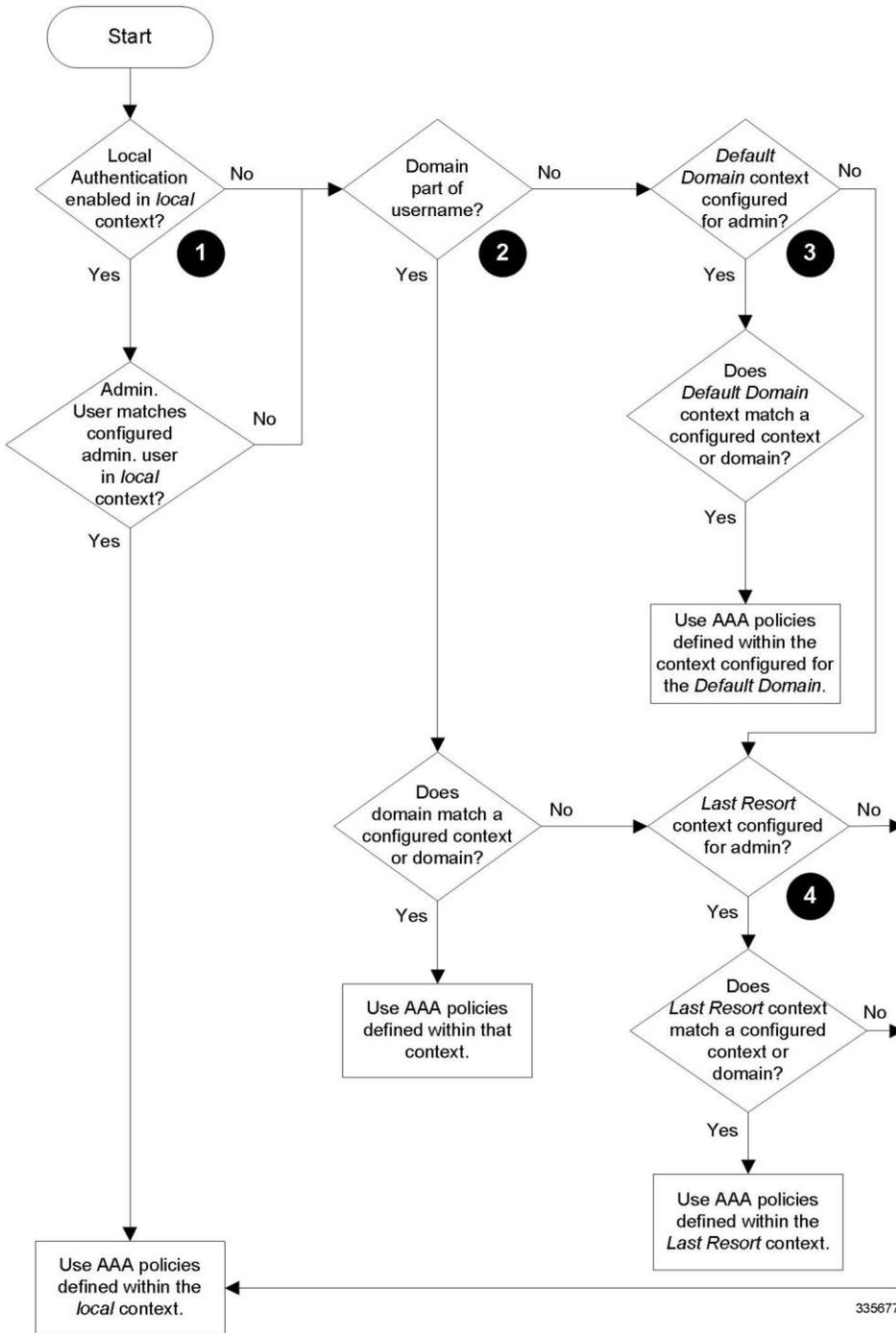


Table 1. Context-level Administrative User AAA Context Selection

Item	Description
1	<p>During authentication, the system determines whether local authentication is enabled in the <i>local</i> context.</p> <p>If it is, the system attempts to authenticate the administrative user in the <i>local</i> context. If it is not, proceed to item 2 in this table.</p> <p>If the administrative user's username is configured, authentication is performed by using the AAA configuration within the <i>local</i> context. If not, proceed to item 2 in this table.</p>
2	<p>If local authentication is disabled on the system or if the administrative user's username is not configured in the <i>local</i> context, the system determines if a domain was received as part of the username.</p> <p>If there is a domain and it matches the name of a configured context or domain, the systems uses the AAA configuration within that context.</p> <p>If there is a domain and it does not match the name of a configured context or domain, Go to item 4 in this table.</p> <p>If there is no domain as part of the username, go to item 3 in this table.</p>
3	<p>If there was no domain specified in the username or the domain is not recognized, the system determines whether an <i>AAA Administrator Default Domain</i> is configured.</p> <p>If the default domain is configured and it matches a configured context, the AAA configuration within the <i>AAA Administrator Default Domain</i> context is used.</p> <p>If the default domain is not configured or does not match a configured context or domain, go to item 4 item below.</p>
4	<p>If a domain was specified as part of the username but it did not match a configured context, or if a domain was not specified as part of the username, the system determines if the <i>AAA Administrator Last Resort context parameter</i> is configured.</p> <p>If a last resort, context is configured and it matches a configured context, the AAA configuration within that context is used.</p> <p>If a last resort context is not configured or does not match a configured context or domain, the AAA configuration within the <i>local</i> context is used.</p>

Context Selection for Subscriber Sessions

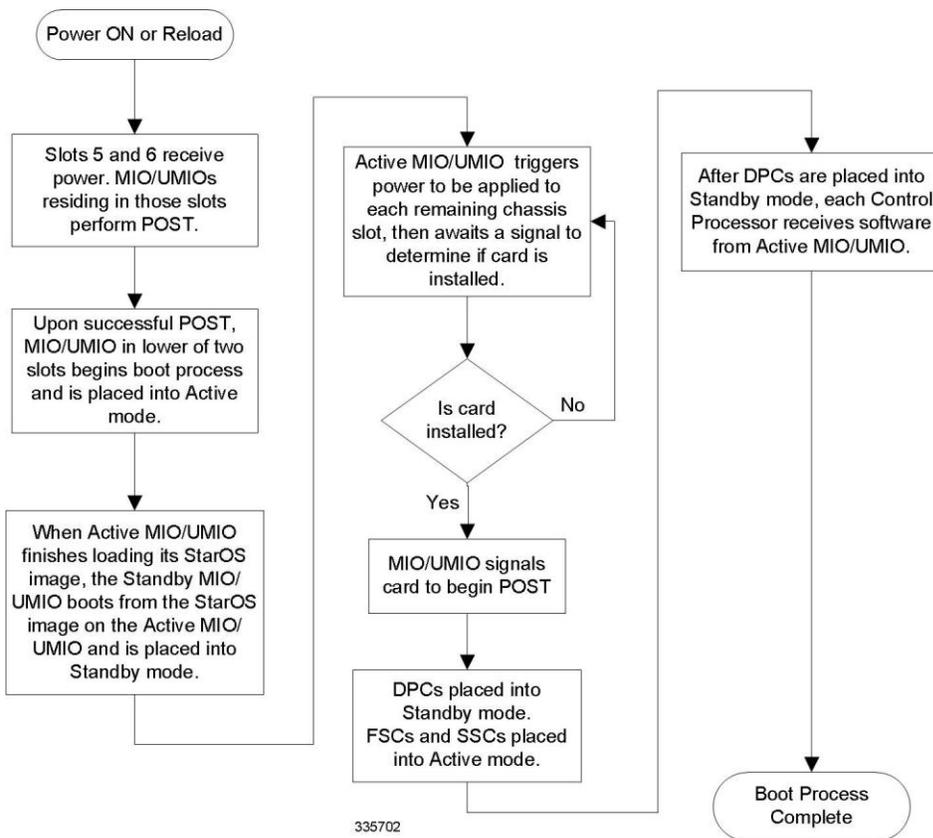
The context selection process for a subscriber session is more involved than that for the administrative users. Subscriber session context selection information for specific products is located in the *Administration Guide* for the individual product.

Understanding the ASR 5500 Boot Process

Part of the configuration process requires that you allocate hardware resources for processing and redundancy. Therefore, before you configure the system, it is important to understand the boot process which determines how the hardware components are brought on line.

The following flowchart shows each step in the startup process. For additional information about system configuration files, refer to the *Understanding Configuration Files* section.

Figure 2. ASR 5500 Process Flowchart



The following steps describe the system's boot process:

- Step 1** When power is first applied to the chassis, or after a reboot, only the MIO/UMIOs in slot 5 and slot 6 receive power.
- Step 2** During the startup process, the MIO/UMIO performs a series of power-on self tests (POSTs) to ensure that its hardware is operational.
- Step 3** If the MIO/UMIO in slot 5 successfully executes all POSTs, it becomes the active MIO. The MIO in slot 6 becomes the standby card. If there is a problem with the MIO in slot 5, the MIO in slot 6 becomes the active MIO.
- Step 4** The active MIO/UMIO begins loading the operating system software image designated in the boot stack. The boot stack entries are contained in the boot.sys file that resides on flash memory on the MIO/UMIO. The standby MIO/UMIO

observes the active card startup. If the file on the active MIO/UMIO is loaded normally, the standby MIO/UMIO boots from the active card image. If the active MIO/UMIO experiences problems during this phase, the standby MIO/UMIO loads its software image designated by its own boot stack entry in its boot.sys file and takes over control of the system as the active MIO/UMIO.

- Step 5** After the software image is loaded into its memory, the active MIO/UMIO determines whether other cards are installed in the chassis by applying power to the other chassis slots and signalling them. If the chassis slot contains a card, power is left On to that slot. All empty slots are powered off.



Important: If no MIOs are installed or if both fail to boot successfully, no other card installed in the system will boot.

- Step 6** When power is applied to the DPC/UDPCs installed in the system, they each perform their own series of POSTs.
- Step 7** After successful POST, each DPC/UDPC enters standby mode.
- Step 8** After entering the standby mode, each of the control processors (CPs) on the DPC/UDPC communicate with the active MIO/UMIO to receive the appropriate code.
- Step 9** Upon successful loading of the software image, the system loads a configuration file designated in the boot stack (boot.sys file). If this is the first time the system is powered on and there is no configuration file, the active MIO/UMIO invokes the system's Quick Setup wizard. Use the Quick Setup wizard to configure basic system parameters for communication across the management network.

The wizard creates a configuration file (system.cfg) that you can use as a starting point for subsequent configurations. This allows you to configure the system automatically by applying the configuration file during any subsequent boot. For additional information about system configuration files, refer to the *Understanding Configuration Files* section.

Understanding Configuration Files

The system supports the use of a file or script to modify configurable parameters. Using a file for offline system configuration reduces the time it takes to configure parameters on multiple systems.

A system configuration file is an ASCII text file that contains commands and configuration parameters. When you apply the configuration file, the system parses through the file line-by-line, testing the syntax and executing the command. If the syntax is incorrect, a message is displayed to the CLI and the system proceeds to the next command. Lines that begin with # are considered remarks and are ignored.

 **Important:** Pipes (|), used with the **grep** and **more** keywords, can potentially cause errors in configuration file processing. Therefore, the system automatically ignores keywords with pipes during processing.

 **Important:** Always save configuration files in UNIX format. Failure to do so can result in errors that prevent configuration file processing.

The commands and configuration data within the file are organized and formatted just as they would be if they were being entered at the CLI prompt. For example, if you wanted to create a context called *source* in the CLI, you would enter the following commands at their respective prompts:

```
[local]host_name# config
[local]host_name(config)# context source
[source]host_name(config-ctx)# end
```

To create a context called *source* using a configuration file, you would use a text editor to create a new file that consists of the following:

```
config

context source

end
```

There are several important things to consider when using configuration files:

- The system automatically applies a configuration file at the end of the boot process. After the system boots up for the first time, a configuration file that you have created and that is tailored to your network needs, can be applied. To make the system use your configuration file, modify the system's boot parameters according to the instructions located in *Software Management Operations*.
- In addition to being applied during the boot process, you can also apply configuration files manually at any time by executing the appropriate commands at the CLI prompt. Refer to the instructions in *Software Management Operations*.

 **Important:** When you apply a configuration file after the boot process, the file does not delete the configuration loaded as part of the boot process. Only those commands that are duplicated are overwritten.

- Configuration files can be stored in any of the following locations:
 - **USB Memory Stick:** Supported via a USB port on the active MIO (/usb1).
 - **Network Server:** Any workstation or server on the network that the system can access using the Secure File Transfer Protocol (SFTP). This is recommended for large network deployments in which multiple systems require the same configuration.
 - **/flash:** a solid-state device with limited storage.
 - **/hd-raid:** internal RAID storage.
- Each time you save configuration changes you made during a CLI session, you can save those settings to a file which you can use as a configuration file.

IP Address Notation

When configuring a port interface via the CLI you must enter an IP address. The CLI always accepts an IPv4 address, and in some cases accepts an IPv6 address as an alternative.

For some configuration commands, the CLI also accepts CIDR notation. Always view the online Help for the CLI command to verify acceptable forms of IP address notation.

IPv4 Dotted-Decimal Notation

An Internet Protocol Version 4 (IPv4) address consists of 32 bits divided into four octets. These four octets are written in decimal numbers, ranging from 0 to 255, and are concatenated as a character string with full stop delimiters (dots) between each number.

For example, the address of the loopback interface, usually assigned the host name localhost, is 127.0.0.1. It consists of the four binary octets 01111111, 00000000, 00000000, and 00000001, forming the full 32-bit address.

IPv4 allows 32 bits for an Internet Protocol address and can, therefore, support 2^{32} (4,294,967,296) addresses

IPv6 Colon-Separated-Hexadecimal Notation

An Internet Protocol Version 6 (IPv6) address has two logical parts: a 64-bit network prefix, and a 64-bit host address part. An IPv6 address is represented by eight groups of 16-bit hexadecimal values separated by colons (:).

A typical example of a full IPv6 address is 2001:0db8:85a3:0000:0000:8a2e:0370:7334

The hexadecimal digits are case-insensitive.

The 128-bit IPv6 address can be abbreviated with the following rules:

- Leading zeroes within a 16-bit value may be omitted. For example, the address fe80:0000:0000:0202:b3ff:fe1e:8329 may be written as fe80:0:0:202:b3ff:fe1e:8329
- One group of consecutive zeroes within an address may be replaced by a double colon. For example, fe80:0:0:0:202:b3ff:fe1e:8329 becomes fe80::202:b3ff:fe1e:8329

IPv6 allows 128 bits for an Internet Protocol address and can support 2^{128} (340,282,366,920,938,000,000,000,000,000,000,000) internet addresses.

CIDR Notation

Classless Inter-Domain Routing (CIDR) notation is a compact specification of an Internet Protocol address and its associated routing prefix. It is used for both IPv4 and IPv6 addressing in networking architectures.

CIDR is a bitwise, prefix-based standard for the interpretation of IP addresses. It facilitates routing by allowing blocks of addresses to be grouped into single routing table entries. These groups (CIDR blocks) share an initial sequence of bits in the binary representation of their IP addresses.

CIDR notation is constructed from the IP address and the prefix size, the latter being the number of leading 1 bits of the routing prefix. The IP address is expressed according to the standards of IPv4 or IPv6. It is followed by a separator character, the slash (/) character, and the prefix size expressed as a decimal number.

The address may denote a single, distinct, interface address or the beginning address of an entire network. In the latter case the CIDR notation specifies the address block allocation of the network. The maximum size of the network is given by the number of addresses that are possible with the remaining, least-significant bits below the prefix. This is often called the host identifier.

For example:

- the address specification 192.168.100.1/24 represents the given IPv4 address and its associated routing prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0.
- the IPv4 block 192.168.0.0/22 represents the 1024 IPv4 addresses from 192.168.0.0 to 192.168.3.255.
- the IPv6 block 2001:DB8::/48 represents the IPv6 addresses from 2001:DB8:0:0:0:0:0:0 to 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF.
- ::1/128 represents the IPv6 loopback address. Its prefix size is 128, the size of the address itself, indicating that this facility consists of only this one address.

The number of addresses of a subnet defined by the mask or prefix can be calculated as 2^n , in which the address size for IPv4 is 32 and for IPv6 is 128. For example, in IPv4, a mask of /29 gives 8 addresses.

Alphanumeric Strings

Some CLI commands require the entry of an alphanumeric string to define a value. The string is a contiguous collection of alphanumeric characters with a defined minimum and maximum length (number of characters).

Character Set

The alphanumeric character set is a combination of alphabetic (Latin letters) and/or numeric (Arabic digits) characters. The set consists of the numbers 0 to 9, letters A to Z (uppercase) and a to z (lowercase). The underscore character (`_`) and dash/hyphen (`-`) are also considered to be members of the alphanumeric set of characters.

Blank spaces (whitespaces or `SPACE` characters) should mostly be avoided in alphanumeric strings, except in certain ruledef formats, such as time/date stamps.

Do not use any of the following “special” characters in an alphanumeric string except as noted below:

- `&` (ampersand)
- `'` (apostrophe)
- `<>` (arrow brackets) [see exception below]
- `*` (asterisk) [see wildcard exception below]
- `{ }` (braces)
- `[]` (brackets)
- `$` (dollar sign) [see wildcard exception below]
- `!` (exclamation point) [see exception below]
- `()` [parentheses]
- `%` (percent) [see exception below]
- `#` (pound sign) [see exception below]
- `?` (question mark)
- `'` (quotation mark – single)
- `"` (quotation mark – double)
- `;` (semicolon)
- `\` (slash – backward) [see exception below]
- `/` (slash – forward) [see exception below]
- `~` (tilde)
- `|` (vertical bar) [see exception below]

The following characters may appear in strings entered in ruledefs, APNs, license keys and other configuration/display parameters:

- `<>` (arrow brackets) [less than or greater than]
- `*` (asterisk) [wildcard]
- `:` (colon)
- `$` (dollar sign) [wildcard]

- . (dot)
- = (equals sign)
- ! (exclamation point)
- % (percent)
- / (slash – forward)
- | (vertical bar)

The following characters may be used to delimit the domain from the user name for global AAA functions:

- @ (at sign)
- - (dash or hyphen)
- # (hash or pound sign)
- % [percent]
- \ (slash – backward) [must be entered as double slash “\\”]
- / (slash – forward)

Quoted Strings

If descriptive text requires the use of spaces between words, the string must be entered within double quotation marks (“”). For example:

```
interface "Rack 3 Chassis 1 port 5/2"
```


Chapter 2

Getting Started

Following successful installation of the system hardware, you must configure a set of software parameters. You then save these settings in a system configuration file that is launched whenever the system is reloaded.

This segment provides instructions for connecting to the console port and creating the initial *local* context management configuration. It includes the following sections:

- [ASR 5500 Configuration](#)
- [Using the ASR 5500 Quick Setup Wizard](#)
- [Using the CLI for Initial Configuration](#)
- [Configuring the System for Remote Access](#)
- [Configuring the Management Interface with a Second IP Address](#)

ASR 5500 Configuration

The first time power is applied to the system, the active Management Input/Output (MIO) card (typically the one installed in chassis slot 5) automatically launches a Quick Setup Wizard on its console port. This wizard guides you through the initial configuration of the system.

The serial console port (logical port 3) is located on the front panel of the MIO card.

You can choose not to use the wizard and perform the initial configuration by issuing commands via the command line interface (CLI). You can manually launch the wizard by running the **setup** command in the Exec mode. Refer to the *Command Line Interface Reference* for details.

The following sections describe how to configure the system.

Using the ASR 5500 Quick Setup Wizard

The Quick Setup Wizard consists of three parts:

- Configuring a context-level security administrator and hostname
- Configuring the Ethernet interface for out-of-band (OOB) management
- Configuring the system for remote CLI access via Telnet, Secure Shell (SSH), or File Transfer Protocol (FTP)

The following figure and table provides a flow diagram that shows the run logic of the wizard along with additional information and notes.

Figure 3. ASR 5500 Quick Setup Wizard Logic Diagram

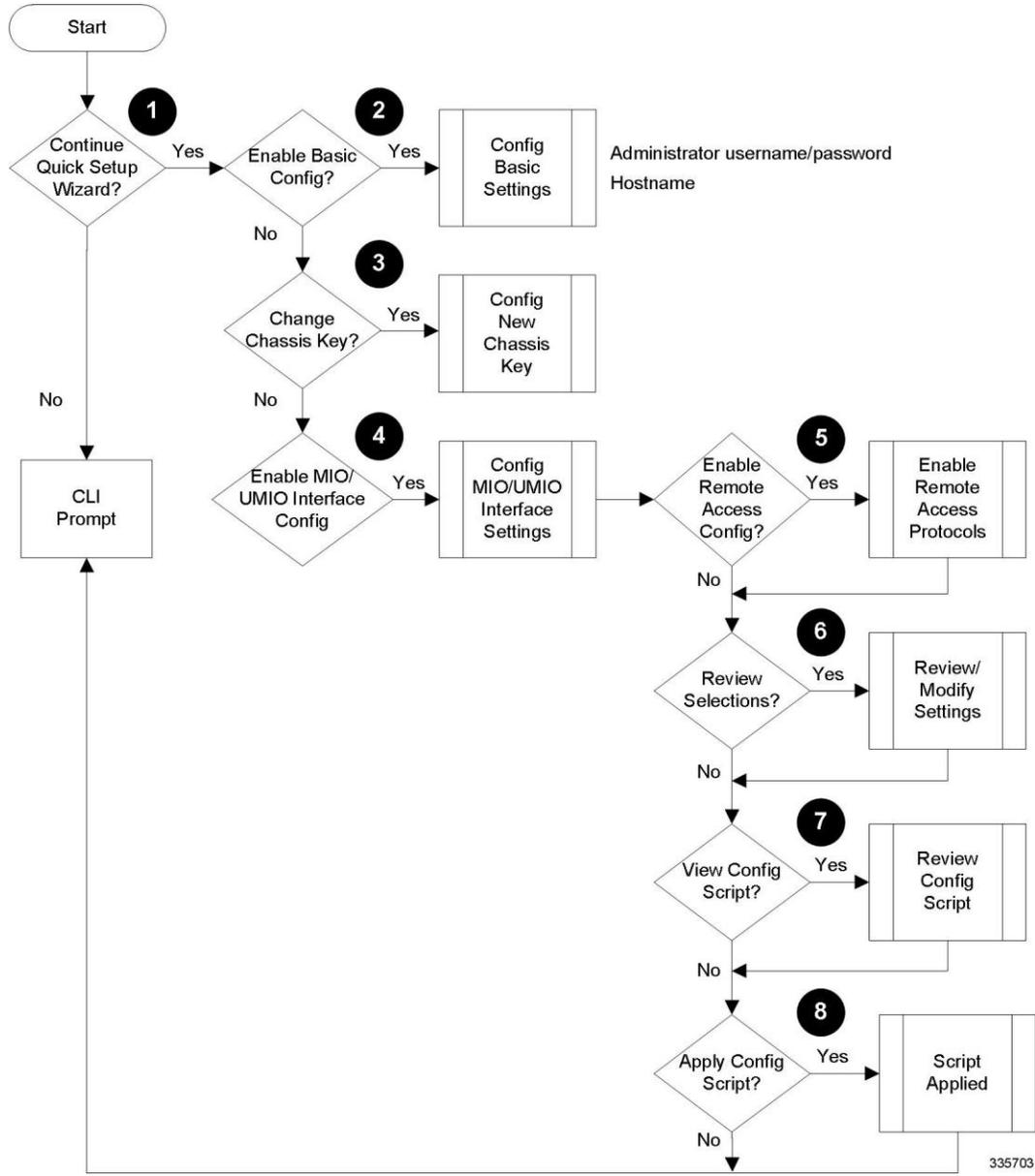
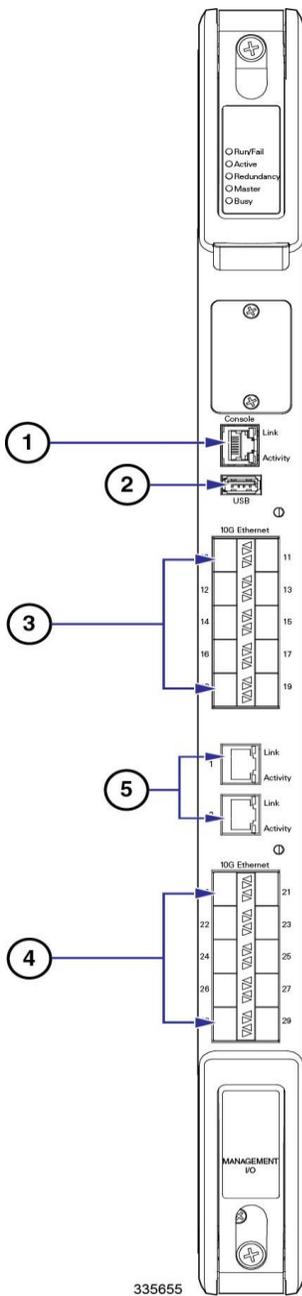


Table 2. Quick Setup Wizard Logic Diagram Callout Descriptions

Figure Item	Task	Description/Notes
1	Enter or exit the wizard.	Enter no at the prompt to automatically be directed to the command line interface (CLI). Proceed to the Using the CLI for Initial Configuration section for instructions on performing an initial system configuration with the CLI.
		Enter setup at the command prompt to re-invoke the wizard.
2	Configure an administrative username/password and the a hostname for the system.	The name of the default administrative user configured through the wizard is <i>admin</i> .
		Administrative user name is an alphanumeric string of 1 through 32 characters that is case sensitive.
		Administrative user password is an alphanumeric string of 1 through 63 characters that is case sensitive.
		Configure a valid, non-null hostname. The hostname is an alphanumeric string of 1 through 63 characters that is case sensitive.
3	Change chassis key value.	<p>A unique chassis key is configured at the factory for each system. This key is used to decrypt encrypted passwords found in generated configuration files. The system administrator can create a unique chassis key that will be used to encrypt passwords stored in configuration files.</p> <p>Enter yes to set a new chassis key. Refer to the instructions in <i>System Settings</i>. Additional information can be found in <i>System Security</i>.</p>
4	Configure a single Management Input/Output (MIO) out-of-band management interface for out-of-band system management.	Traffic on the management LAN is not transferred over the same media as user data and control signaling.
		For security reasons, it is recommended that management functions be maintained on a separate network from user data and control signaling.
		MIO port 1 (mio1) is the 1000Base-T default management port. MIO port 2 (mio2) is available as a secondary management port.
		Use the RJ-45 interfaces to connect the system to the management network with CAT5 Ethernet cable.
		Configure an IP address, subnet mask, and gateway for the interface.
		Instructions for configuring the second management interface on the MIO can be found in <i>System Settings</i> .

Figure Item	Task	Description/Notes
5	Enable various remote access protocols for accessing the system.	Secure Shell (SSH) uses TCP port number 22 by default, if enabled. SSH V1 and/or V2 are supported. If SSH is enabled, you can also enable SSH File Transfer Protocol (SFTP) server functionality.
		Telnet uses TCP port number 23 by default, if enabled.
		The File Transfer Protocol (FTP) uses TCP port number 21 by default, if enabled.
		For maximum security, use only SSH v2.
6	Review and/or modify the configuration of previous prompts.	<ol style="list-style-type: none"> 1. Enter the number of the prompt to be modified. 2. Configure the parameter. 3. <i>Optional.</i> Repeat <i>step 1</i> and <i>step 2</i> to modify additional settings. 4. Enter “done” when you have completed all changes.
7	Review the configure script created by the wizard based on your inputs.	An example of a created script is displayed in the example below. Variables are displayed in italics (<i>variable</i>).
8	Apply the configuration file to the system.	Once applied, the parameter configuration is automatically saved to the system.cfg file stored in MIO flash memory.

Figure 4. MIO Interfaces



1	Console port	2	USB port
3	10 GbE ports, DC-1	4	10 GbE ports, DC-2
5	1 GbE ports (1000Base-T)		

```
config
  system hostname <hostname>
  context local
    administrator <admin_name> password <passwd>
    interface miol
      ip address <ip_address> subnet
      #exit
    ip route 0.0.0.0 0.0.0.0 <gw_address> miol
    ssh key <v1_key>
    ssh key <v2_rsa_key>
    ssh key <v2_dsa_key>
    server sshd
    subsystem sftp
    #exit
    no server telnetd
    server ftpd
    no server telnetd
    #exit
  port ethernet 5/1
  bind interface miol local
  no shutdown
#exit
end
```



Important: Once configuration using the wizard is complete, proceed to instructions on configuring other system parameters.

Using the CLI for Initial Configuration

The initial configuration consists of the following:

- Configuring a context-level security administrator and hostname
- Configuring the Ethernet interface on the MIO
- Configuring the system for remote CLI access via Telnet, SSH, or FTP (secured or unsecured)

This section provides instructions for performing these tasks using the CLI.

Step 1 At the CLI prompt, enter:

```
[local]host_name# configure
[local]host_name(config)#
```

Step 2 Enter the context configuration mode by entering the following command:

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

The *local* context is the system's management context. Contexts allow you to logically group services or interfaces. A single context can consist of multiple services and can be bound to multiple interfaces.

Step 3 Enter the following command to configure a context-level security administrator for the system:

```
administrator name { password password | encrypted password enc_password } [ ftp
] [ no-cli ] [ timeout-absolute absolute_time ] [ timeout-idle idle_time ]
```

Keyword/Variable	Description
<i>name</i>	Specifies the security administrator's name as an alphanumeric string of 1 through 32 characters that is case sensitive.
password <i>password</i>	Specifies the password for the security administrator as an alphanumeric string of 1 through 63 characters that is case sensitive.
encrypted password	Specifies the encrypted password for the security administrator. The keyword is only used by the system when you save configuration scripts. The system displays the encrypted keyword in the configuration file as a flag indicating that the variable following the keyword is the encrypted version of the plain text password. Only the encrypted password is saved as part of the configuration file.
ftp	Specifies that the security administrator is allowed to access the system with the File Transfer Protocol (FTP). This option is useful to upload files (configuration or software images) to the system's flash memory.
no-cli	Specifies that the security administrator cannot access the system command line interface (CLI). Use this keyword in conjunction with the ftp keyword to allow access to the system with FTP only.
timeout-absolute	Specifies the maximum amount of time (in seconds) that the operator can maintain a session with the system, as an integer from 0 through 300000000. Default: 0 If the absolute timeout value is exceeded, the operator session is automatically terminated.
timeout-idle	Specifies the maximum amount of time (in seconds) that an operator session can remain idle before being automatically terminated. This is an integer from 0 through 300000000. Default: 0.



Important: You must configure a context-level security administrator during the initial configuration. After you complete the initial configuration process and end the CLI session, if you have not configured a security administrator, CLI access will be locked.

Step 4 Enter the following command at the prompt to exit the context configuration mode:

```
[local]host_name(config-ctx) # exit
[local]host_name(config) #
```

Step 5 Enter the following command to configure a hostname by which the system will be recognized on the network:

```
[local]host_name(config) # system hostname host_name
```

host_name is the name by which the system will be recognized on the network. The hostname is an alphanumeric string of 1 through 63 characters that is case sensitive.

Step 6 Configure the network interfaces on the MIO using the following instructions:

Step a Enter the context configuration mode by entering the following commands:

```
[local]host_name(config) # context local
[local]host_name(config-ctx) #
```

Step b Enter the following command to specify a name for the interface:

```
[local]host_name(config-ctx) # interface interface_name
```

interface_name is the name of the interface expressed as an alphanumeric string of 1 through 79 characters that is case sensitive. The following prompt appears as the system enters the Ethernet Interface Configuration mode:

```
[local]host_name(config-if-eth) #
```

Step c Configure an IP address for the interface configured in the previous step by entering the following command:

```
{ ip address | ipv6 address } ipaddress subnetmask
```

Variable	Description
IPv4 – ip address	
<i>ipaddress</i>	Specifies the IP address for the interface in IPv4 dotted-decimal format.
<i>subnetmask</i>	Specifies the subnet mask for the interface in IPv4 dotted-decimal or CIDR notation.
IPv6 – ipv6 address	
<i>ipaddress</i>	Specifies the IP address for the interface in IPv6 colon-separated-hexadecimal format.
<i>subnetmask</i>	Specifies the subnet mask for the interface in IPv6 colon-separated-hexadecimal or CIDR notation.



Important: If you are executing this command to correct an address or subnet that was mis-configured with the Quick Setup Wizard, you must verify the default route and port binding configuration. Use *step 11* and *step 6* of this procedure. If there are issues, perform steps *7e* through *7k* to reconfigure the information.

Step d Enter the following command to exit the Ethernet interface configuration mode:

```
[local]host_name(config-if-eth)# exit
[local]host_name(config-ctx)#
```

Step e Configure a static route, if required, to point the system to a default gateway. Entering the following command:

```
{ ip | ipv6 } route gw_address interface_name
```

Variable	Description
<i>gw_address</i>	Specifies the IP address of the default gateway in IPv4 dotted-decimal format or IPv6 colon-separated-hexadecimal format.
<i>interface_name</i>	Specifies the name of the interface that was configured in step 7b.

Step f Enter the following to exit from the context configuration mode:

```
[local]host_name(config-ctx)# exit
[local]host_name(config)#
```

Step g Enter the Ethernet Port Configuration mode:

```
port ethernet slot#/port#
```

Variable	Description
<i>slot#</i>	The actual chassis slot in which the MIO card is installed. This could be either slot number 5 or 6.
<i>port#</i>	The physical port on the MIO that will be used. This will be either port 1 or 2. Port 1 represents the top most port.

Step h Bind the port to the interface that you created in step 7b. Binding associates the port and all of its settings to the interface. Enter the following command:

```
[local]host_name(config-port-<slot#/port#>)# bind interface interface_name local
[local]host_name(config-port-<slot#/port#>)# no shutdown
```

interface_name is the name of the interface that you configured in step 7b.

Step i Exit the Ethernet Interface Configuration mode by entering the command:

```
[local]host_name(config-port-<slot#/port#>)# exit
[local]host_name(config)#
```

 **Important:** Refer below for instructions on configuring the MIO management interface with a second IP address.

Configuring the System for Remote Access

Configure the system for remote access. An administrative user may access the system from a remote location over a local area network (LAN) or wide area network (WAN):

- Telnet
- Secure Shell (SSH)
- File Transfer Protocol (FTP) (secured or unsecured)
- Trivial File Transfer Protocol (TFTP)

 **Important:** If there are two simultaneous telnet sessions, and one administrator deletes the context into which the other administrator is logged, the administrator in the deleted context will not be automatically kicked into the *local* context. Although the deleted context will still appear in the CLI prompt, context specific commands will generate errors.

 **Important:** For maximum security, use SSH v2.

Step 1 Enter the context configuration mode by entering the following command:

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

Step 2 Configure the system to allow Telnet access, if desired:

```
[local]host_name(config-ctx)# server telnetd
```

Step 3 Configure the system to allow SSH access, if desired:

```
[local]host_name(config-ctx)# ssh generate key [ type { v1-rsa | v2-rsa | v2-dsa
} ]
```

 **Important:** *v2-rsa* is the recommended key type.

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)# subsystem sftp
[local]host_name(config-sshd)# exit
```

Step 4 Configure the system to allow FTP access, if desired, by entering the following command:

```
[local]host_name(config-ctx)# server ftpd
```

Step 5 Exit the configuration mode by entering the following command:

```
[local]host_name(config-ctx)# end
[local]host_name#
```

Step 6 Verify the configuration by entering the following command:

```
[local]host_name# show configuration
```

The CLI output should be similar to the sample output:

```
context local
  interface <interface_name>
    ip address <ipaddress> <subnetmask>
  exit
  subscriber default
  exit
  administrator <admin_name> password <admin_password>
  server telnetd
  server ftpd
  ssh generate key
  server sshd
  subsystem sftp
  exit
port ethernet 5/1
  bind interface <interface_name> local
  exit
port ethernet 5/1
  no shutdown
  exit
snmp engine-id local 800007e580ed826c191ded2d3d
end
```

Step 7 Verify the configuration of the IP routes by entering the following command:

```
[local]host_name# show ip route
```

The CLI output should be similar to the sample output:

```
"*" indicates the Best or Used route.
Destination      Nexthop          Protocol  Prec Cost Interface
*0.0.0.0/0       <ipaddress>     static    1    0    miol
*<network>       0.0.0.0         connected 0    0    miol
```

Step 8 Verify the interface binding by entering the following command:

```
[local]host_name# show ip interface name <interface_name>
```

<interface_name> is the name of the interface that was configured in *step 7b*. The CLI output should be similar to the sample output:

```
Intf Name:      miolIntf Type:      Broadcast
Description:
IP State:       UP (Bound to 5/1 untagged, ifIndex 83951617)
IP Address:     <ipaddress>       Subnet Mask:      <subnetmask>
Bcast Address: <bcastaddress>   MTU:              1500
Resoln Type:   ARP             ARP timeout:      3600 secs
Number of Secondary Addresses: 0
```

Step 9 Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring the Management Interface with a Second IP Address

If necessary, you can configure a second IP address on the MIO management interface.

Step 1 Enter the configuration mode by entering the following command at the prompt:

```
[local]host_name# configure  
[local]host_name(config)#
```

Step 2 Enter the following to enter the context configuration mode:

```
[local]host_name(config)# context local  
[local]host_name(config-ctx)#
```

Step 3 Enter the interface slot number and port number by entering the following command:

```
[local]host_name(config-ctx)# 5/1  
[local]host_name(config-if-eth)#
```

Step 4 Enter the secondary IP address and subnet mask by entering the following command:

```
[local]host_name(config-if-eth)# { ip | ipv } address ipaddress subnet_mask  
secondary
```

Step 5 Exit the configuration mode by entering the following command:

```
[local]host_name(config-if-eth)# end
```

Step 6 Confirm the interface ip addresses by entering the following command:

```
[local]host_name# show config context local
```

The CLI output should look similar to this example:

```
config  
  context local  
    interface <interface_name>  
      ip address <ipaddress> <subnetmask>  
      ip address <ipaddress> <subnetmask> secondary  
    #exit
```

Step 7 Save your configuration as described in *Verifying and Saving Your Configuration*.

Chapter 3

System Settings

This segment provides instructions for configuring the following system options:

- [Configuring a Second Management Interface](#)
- [Configuring System Timing](#)
- [Enabling CLI Timestamping](#)
- [Configuring System Administrative Users](#)
- [Configuring TACACS+ for System Administrative Users](#)
- [Configuring a Chassis Key](#)
- [Configuring MIO/UMIO Port Redundancy](#)
- [Configuring Data Processing Card \(DPC\) Availability](#)
- [Configuring ASR 5500 Link Aggregation](#)

It is assumed that the procedures to initially configure the system as described in *Getting Started* have been completed.

 **Important:** The commands used in the configuration examples in this section are the most likely-used commands and/or keyword options. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

Configuring a Second Management Interface

Refer to *Getting Started* for instructions on configuring a system management interface on the Management Input/Output (MIO) or Management Input/Output Universal (UMIO) card. This section provides described how to configure a second management interface.

Use the following example to configure a second management interface:

```
configure

context local

    interface <interface_name>

        ip address <ipaddress subnetmask>

        exit

    ip route 0.0.0.0 0.0.0.0 <gw_address interface_name>

    exit

port ethernet <slot#/port#>

    bind interface <interface_name> local

    no shutdown

    media [ rj45 | sfp ]

end
```

Notes:

- For **port ethernet slot#**, use the actual chassis slot in which the active MIO/UMIO is installed (slot number 5 or 6).
- Enter IP addresses using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- For **port ethernet port#**, use the physical port on the MIO/UMIO card that will be used. This is either port 1 or 2. Port 1 represents the top-most port (either RJ-45 or SFP).
- The MIO/UMIO is equipped with RJ-45 (1000Base-T copper) interfaces. The RJ-45 interfaces connect the system to the management network with CAT3 or CAT5 Ethernet cable.
- *Option:* In the Ethernet Port configuration mode, configure the port speed, if needed, by entering the **medium** command. Refer to the *Command Line Interface Reference* for a complete explanation of this command.
- In the { **ip | ipv6** } **route** command, other keyword options, instead of the gateway IP address, are available and include: **next-hop** IP address, **point-to-point**, and **tunnel**.

Verifying and Saving Your Interface and Port Configuration

Verify that your interface configuration settings are correct by entering the following command:

```
show ip interface
```

The output from this command should be similar to that shown below. In this example an interface named *mgmt2* was configured in the local context.

```
Intf Name:      mgmt2
Intf Type:      Broadcast
Description:    management2
VRF:           None
IP State:       UP (Bound to 5/2)
IP Address:     192.168.100.3      Subnet Mask:    255.255.255.0
Bcast Address:  192.168.100.255     MTU:           1500
Resoln Type:    ARP              ARP timeout:    60 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
```

Verify that the port configuration settings are correct by entering the following command:

```
show configuration port <slot#/port#>
```

slot# is the chassis slot number of the line card where the physical port resides. *slot#* is either 5 or 6. *port#* is the number of the port (either 1 or 2).

This following command produces an output similar to the one shown below. It displays the configuration of port 2 of the MIO/UMIO installed in chassis slot 5. In this example, the port is bound to an interface called *mgmt2*.

```
config
  port ethernet 5/2
    description management2
    no shutdown
    bind interface mgmt2 local
  #exit
end
```

Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring System Timing

The system is equipped with a clock that supplies the timestamp for statistical counters, accounting records, logging, and event notification. After the initial configuration of the system clock, you can configure the system to communicate with one or more Network Time Protocol (NTP) server(s) to ensure that the clock is always accurate.

In the event of a power outage, the clock is maintained with an accuracy of +/- one minute per month for up to 10 years. This ensures that when power is restored, the system is ready to process sessions and generate accounting, log, and event data with accurate timestamps.

In addition to configuring the timing source, you must configure the system's time zone.

Setting the System Clock and Time Zone

Use the following command example to configure the system clock and time zone:

```
clock set <date:time>

configure

    clock timezone <timezone> [ local ]

end
```

Notes:

- Enter the date and time in the format YYYY:MM:DD:HH:mm or YYYY:MM:DD:HH:mm:ss.
- Refer to the online Help for the **clock timezone** command for a complete list of supported time zones.
- The optional **local** keyword indicates that the time zone specified is the local timezone.
- Daylight Savings Time is automatically adjusted for time zones supporting it.

Save your configuration as described in *Verifying and Saving Your Configuration*.

Verifying and Saving Your Clock and Time Zone Configuration

Enter the following command to verify that you configured the time and time zone correctly:

```
show clock
```

The output displays the date, time, and time zone that you configured.

Configuring Network Time Protocol Support

This section provides information and instructions for configuring the system to enable the use of the Network Time Protocol (NTP).

 **Important:** Configure the system clock and time zone prior to implementing NTP support. This greatly reduces the time period that must be corrected by the NTP server.

Many of the services offered by the ASR 5x00 platform require accurate timekeeping derived through NTP. If the time reference(s) used by StarOS are not accurate, the services may be unreliable. For this reason it should be assumed that normal system operation requires that NTP be configured.

The system uses NTP to synchronize internal clocks on the chassis to external time sources (typically GPS NTP sources, or other Stratum 2 or 3 servers, switches or routers).

By default, NTP is not enabled externally and should be configured when the system is initially installed. When enabled, the active MIO/UMIO will synchronize with external sources. If not enabled, the active MIO/UMIO will use its local clock as a time source. In the event of an NTP server or network outage, an already running MIO/UMIO will continue to use NTP to maintain time accuracy, but in a holdover mode.

All cards with CPUs synchronize to the active MIO/UMIO internally. This occurs even if an external NTP server is not configured. In the event of a MIO/UMIO switchover, all other cards will start synchronizing with the newly active MIO/UMIO automatically.

The system should have:

- NTP enabled.
- NTP configured for use in the local context only. Use of other contexts (which can be specified in the enable configurable) will cause issues.
- NTP configured for at least three external NTP servers. With three or more servers, outliers and broken or misconfigured servers can be detected and excluded. Generally, the more servers the better (within reason).

 **Important:** Do not configure any external NTP servers using the **prefer** keyword. The NTP clock selection algorithms already have the built-in ability to pick the best server. Use of **prefer** usually results in a poorer choice than NTP can determine for itself.

 **Important:** Do not change the **maxpoll**, **minpoll**, or **version** keyword settings unless instructed to do so by Cisco TAC.

Use the following example to configure the necessary NTP association parameters:

```
configure
 ntp
  enable
  server <ip_address1>
  server <ip_address2>
```

```
server <ip_address3>  
  
end
```

Notes:

- By default *context_name* is set to *local*. This is the recommended configuration.
- A number of options exist for the **server** command. Refer to *NTP Configuration Mode Commands* in the *Command Line Interface Reference* for more information.
- Enter the IP address of NTP servers using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.



Important: Configure the system with at least three (preferably four) NTP servers.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring NTP Servers with Local Sources

NTP can use network peers, local external clocks (such as GPS devices), or a local clock with no external source.

A local clock with no external source is usually a last-resort clock when no better clock is available. It is typically configured on a site's intermediate NTP server so that when a WAN network outage occurs, hosts within the site can continue to synchronize amongst themselves.

You can configure this in `ntpd` or on many commercially available NTP devices. This local clock should always have a high stratum number (8+) so that under normal conditions (when real sources are available) this local clock will not be used.

Using a Load Balancer

The NTP daemon and protocol assume that each configured server is running NTP. If a NTP client is configured to synchronize to a load balancer that relays and distributes packets to a set of real NTP servers, the load balancer may distribute those packets dynamically and confuse the NTP client. NTP packets are latency and jitter sensitive. Relaying them through a load balancer can confuse the NTP client and is not a supported practice.

Verifying the NTP Configuration

Verify the NTP configuration is correct. Enter the following command at the Exec mode prompt:

```
show ntp associations
```

The output displays information about all NTP servers. See the output below for an example deploying two NTP servers.

```

+----Peer Selection: ( ) - Rejected / No Response
|                   (x) - False Tick
|                   (.) - Excess
|                   (-) - Outlyer
|                   (+) - Candidate
|                   (#) - Selected
|                   (*) - System Peer
|                   (o) - PPS Peer
v
remote              local          st poll reach  delay  offset  disp
=====
*10.0.4.5           0.0.0.0          3  64    1 0.00060 0.000019 0.00029
*10.0.4.5           0.0.0.0          3  64    1 0.00060 0.000019 0.00029

```

The following table describes the parameters output by the **show ntp associations** command.

Column Title	Description
remote	List of the current NTP servers. One of these characters precedes each IP address to show the server's current condition: <ul style="list-style-type: none"> • () Rejected/No response • X False tick • . Excess • - Outlyer • + Candidate • # Selected • * System peer • (o) PPS peer
refid	Last reported NTP reference to which the server is synchronizing.
st	NTP server stratum level.
t	Communication type: broadcast, multicast, etc.
when	Number of seconds since the last contact.
poll	Polling interval between the system and the NTP server.
reach	Octal value of the reachability shift register indicating which responses were received for the previous eight polls to this NTP server.
delay	Round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.
jitter	Jitter in milliseconds between the system and the NTP server.

Enabling CLI Timestamping

To display a timestamp (date and time) for every command that is executed on the CLI, enter the following command at the root prompt for the Exec mode:

```
timestamps
```

The date and time appear immediately after you execute the command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring System Administrative Users

Getting Started describes how to configure a context-level security administrator for the system.

This section provides instructions for configuring additional administrative users having the following privileges:

- **Security Administrators:** have read-write privileges and can execute all CLI commands, including those available to Administrators, Operators, and Inspectors
- **Administrators:** have read-write privileges and can execute any command in the CLI except for a few security-related commands that can only be configured by Security Administrators. Administrators can configure or modify system settings and execute all system commands, including those available to the Operators and Inspectors.
- **Operators:** have read-only privileges to a larger subset of the Exec Mode commands. They can execute all commands that are part of the inspector mode, plus some system monitoring, statistic, and fault management functions. Operators do not have the ability to enter the Config Mode.
- **Inspectors:** are limited to a few read-only Exec Mode commands. The bulk of these are **show** commands for viewing a variety of statistics and conditions. An Inspector cannot execute **show configuration** commands and does not have the privilege to enter the Config Mode.

Configuration instructions are categorized according to the type of administrative user: context-level or local-user.



Important: For information on the differences between these user privileges and types, refer to *Getting Started*.

Configuring Context-level Administrative Users

This section contains information and instructions for configuring context-level administrative user types.

Configuring Context-level Security Administrators

Use the example below to configure additional security administrators:

```
configure
  context local
    administrator <name> { password <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **administrator** command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring Context-level Administrators

Use the example below to configure context-level administrators:

```
configure
  context local
    config-administrator <name> { password <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **config-administrator** command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring Context-level Operators

Use the example below to configure context-level operators:

```
configure
  context local
    operator <name> { password <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **operator** command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring Context-level Inspectors

Use the example below to configure context-level inspectors:

```
configure
  context local
    inspector <name> { password <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **inspector** command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Verifying Context-level Administrative User Configuration

Verify that the configuration was successful by entering the following command:

```
show configuration context local
```

This command displays all of the configuration parameters you modified within the Local context during this session. The following displays sample output for this command. In this example, a security administrator named *testadmin* was configured.

```
config
  context local
    interface mgmt1
      ip address 192.168.1.10 255.255.255.0
    #exit
    subscriber default
    #exit
    administrator testadmin encrypted password fd01268373c5da85
    inspector testinspector encrypted password 148661a0bb12cd59
  exit
  port ethernet 24/1
    bind interface mgmt1 local
  exit
end
```

Configuring Local-User Administrative Users

Use the example below to configure local-user administrative users:

```
configure
  local-user username <name>
end
```

Notes:

- Additional keyword options are available identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **local-user username** command.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Verifying Local-User Configuration

Verify that the configuration was successful by entering the following command:

```
show local-user verbose
```

This command displays information on configured local-user administrative users. A sample output for this command appears below. In this example, a local-user named *SAUser* was configured.

```
Username:                SAUser
Auth Level:              secadmin
Last Login:              Never
Login Failures:          0
Password Expired:        Yes
Locked:                  No
Suspended:               No
Lockout on Pw Aging:     Yes
Lockout on Login Fail:   Yes
```

Configuring TACACS+ for System Administrative Users

This section describes TACACS+ (Terminal Access Controller Access Control System+) AAA (Authentication Authorization and Accounting) service functionality and configuration on the ASR 5x00.

Operation

TACACS+ is a secure, encrypted protocol. By remotely accessing TACACS+ servers that are provisioned with the administrative user account database, the ASR 5x00 can provide TACACS+ AAA services for system administrative users. TACACS+ is an enhanced version of the TACACS protocol that uses TCP instead of UDP.

The ASR 5x00 system serves as the TACACS+ Network Access Server (NAS). As the NAS the system requests TACACS+ AAA services on behalf of authorized system administrative users. For the authentication to succeed, the TACACS+ server must be in the same local context and network accessed by the system.

The system supports TACACS+ multiple-connection mode. In multiple-connection mode, a separate and private TCP connection to the TACACS+ server is opened and maintained for each session. When the TACACS+ session ends, the connection to the server is terminated.

TACACS+ is a system-wide function on the ASR 5x00. TACACS+ AAA service configuration is performed in TACACS Configuration Mode. Enabling the TACACS+ function is performed in the Global Configuration Mode. The system supports the configuration of up to three TACACS+ servers.

Once configured and enabled on the system, TACACS+ authentication is attempted first. By default, if TACACS+ authentication fails, the system then attempts to authenticate the user using non-TACACS+ AAA services, such as RADIUS.

User Account Requirements

Before configuring TACACS+ AAA services on the ASR 5x00, note the following TACACS+ server and system user account provisioning requirements:

TACACS+ User Account Requirements

The TACACS+ server must be provisioned with the following TACACS+ user account information:

- A list of known administrative users.
- The plain-text or encrypted password for each user.
- The name of the group to which each user belongs.
- A list of user groups.
- TACACS+ privilege levels and commands that are allowed/denied for each group.

 **Important:** TACACS+ privilege levels are stored as Attribute Value Pairs (AVPs) in the network's TACACS+ server database. Users are restricted to the set of commands associated with their privilege level. A mapping of TACACS+ privilege levels to ASR 5x00 CLI administrative roles and responsibilities is provided in the table below.

Table 3. Mapping of TACACS+ Privilege Levels to CLI Administrative Roles

TACACS+ Privilege Level	CLI Administrative Access Privileges				CLI Role
	CLI	FTP	ECSEMS	Lawful Intercept	
0	Yes	No	No	No	Inspector
1	Yes	No	Yes	No	Inspector
2	No	Yes	No	No	Inspector
3	Yes	Yes	No	No	Inspector
4	Yes	Yes	Yes	No	Inspector
5	Yes	No	No	No	Operator
6	Yes	No	Yes	No	Operator
7	No	Yes	No	No	Operator
8	Yes	Yes	No	No	Operator
9	Yes	Yes	Yes	No	Operator
10	Yes	No	No	No	Administrator
11	Yes	No	Yes	No	Administrator
12	No	Yes	No	No	Administrator
13	Yes	Yes	No	Yes	Administrator
14	Yes	Yes	Yes	No	Administrator
15	Yes	Yes	Yes	Yes	Administrator

User Account Requirements

TACACS+ users who are allowed administrative access to the system must have the following user account information defined on the ASR 5x00:

- username
- password
- administrative role and privileges

 **Important:** For instructions on defining users and administrative privileges on the system, refer to *Configuring System Administrative Users*.

Configuring TACACS+ AAA Services

This section provides an example of how to configure TACACS+ AAA services for administrative users on the system.

 **Caution:** When configuring TACACS+ AAA services for the first time, the administrative user must use non-TACACS+ services to log into the ASR 5x00. Failure to do so will result in the TACACS+ user being denied access to the system.

Log in to the system using non-TACACS+ services.

Use the example below to configure TACACS+ AAA services on the system:

```
configure
    tacacs mode
        server priority <priority_number> ip-address <tacacs+srvr_ip_address>
    end
```

Note:

- **server priority** *priority_number*: Must be a number from 1 to 3, that specifies the order in which this TACACS+ server will be tried for TACACS+ authentication. 1 is the highest priority, and 3 is the lowest.
- **ip-address**: Must be the IPv4 address of a valid TACACS+ server that will be used for authenticating administrative users accessing this system via TACACS+ AAA services.
- By default, the TACACS+ configuration will provide authentication, authorization, and accounting services.

Enable TACACS+ on the ASR 5x00:

```
configure
    aaa tacacs+
end
```

Save the configuration as described in *Verifying and Saving Your Configuration*.

 **Important:** For complete information on all TACACS+ Configuration Mode commands and options, refer to *TACACS Configuration Mode Commands* in the *Command Line Reference*.

Verifying the TACACS+ Configuration

This section describes how to verify the TACACS+ configuration:

Log out of the system CLI, then log back in using TACACS+ services.

 **Important:** Once TACACS+ AAA services are configured and enabled on the ASR 5x00, the system first will try to authenticate the administrative user via TACACS+ AAA services. By default, if TACACS+ authentication fails, the system then continues with authentication using non-TACACS+ AAA services.

At the Exec Mode prompt, enter the following command:

```
show tacacs
```

The command output provides summary information for each active TACACS+ session such as username, login time, login status, current session state and privilege level.

An example of this command's output is provided below. In this example, a system administrative user named *asradmin* has successfully logged in to the system via TACACS+ AAA services.

```
active session #1:
  login username           : asradmin
  login tty                : /dev/pts/1
  time of login           : Fri Oct 22 13:19:11 2011
  login server priority    : 1
  current login status     : pass
  current session state    : user login complete
  current privilege level  : 15
  remote client application : ssh
  remote client ip address : 111.11.11.11
  last server reply status : -1
total TACACS+ sessions    : 1
```

 **Important:** For details on all TACACS+ maintenance commands, refer to the *Command Line Interface Reference*.

Configuring a Chassis Key

A unique chassis key is configured at the factory for each system. This key is used to decrypt encrypted passwords found in generated configuration files. The system administrator can create a unique chassis key that will be used to encrypt passwords stored in configuration files.



Important: The Quick Setup Wizard also prompts the user to enter a chassis key value.

The Exec mode **chassis key value** *key_string* command identifies the chassis which can encrypt and decrypt encrypted passwords in the configuration file. If two or more chassis are configured with the same chassis key value, the encrypted passwords can be decrypted by any of the chassis sharing the same chassis key value. As a corollary to this, a given chassis key value will not be able to decrypt passwords that were encrypted with a different chassis key value.

The *key_string* is an alphanumeric string of 1 through 16 characters. The chassis key is stored as a one-way encrypted value, much like a password. For this reason, the chassis key value is never displayed in plain-text form.

The Exec mode **chassis keycheck** *key_string* command generates a one-way encrypted key value based on the entered *key_string*. The generated encrypted key value is compared against the encrypted key value of the previously entered chassis key value. If the encrypted values match, the command succeeds and keycheck passes. If the comparison fails, a message is displayed indicating that the key check has failed. If the default chassis key (no chassis key) is currently being used, this key check will always fail since there will be no chassis key value to compare against.

Use the **chassis keycheck** command to verify whether multiple chassis share the same chassis key value.



Important: Only a user with Security Administrator or Administrator privilege can execute the **chassis key value** and **chassis keycheck** commands.

For additional information, refer to *Exec Mode Commands (A-C)* in the *Command Line Interface Reference*.

Configuring MIO/UMIO Port Redundancy

Port redundancy for MIOs provides an added level of redundancy that minimizes the impact of network failures that occur external to the system. Examples include switch or router port failures, disconnected or cut cables, or other external faults that cause a link down error.

 **Caution:** To ensure that system card and port-level redundancy mechanisms function properly, disable the Spanning Tree protocol on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

By default, the system provides port-level redundancy when a failure occurs, or you issue the **port switch to** command. In this mode, the ports on active and standby MIO/UMIO cards have the same MAC address, but since only one of these ports may be active at any one time there are no conflicts. This eliminates the need to transfer MAC addresses and send gratuitous ARPs in port failover situations. Instead, for Ethernet ports, three Ethernet broadcast packets containing the source MAC address are sent so that the external network equipment (switch, bridge, or other device) can re-learn the information after the topology change. However, if card removal is detected, the system sends out gratuitous ARPs to the network because of the MAC address change that occurred on the specific port.

With port redundancy, if a failover occurs, only the specific port(s) become active. For example; if port 5/1 fails, then port 6/1 becomes active, while all other active ports on the line card in slot 5 remain in the same active state. In port failover situations, use the **show port table** command to check that ports are active on both cards and that both cards are active.

Take care when administratively disabling a port that is one of a redundant pair. A redundant pair comprises both the active and standby ports—for example 5/1 and 6/1. If 5/1 is active, administratively disabling 5/1 through the CLI does not make 6/1 active. It disables both 5/1 and 6/1 because an action on one port has the same effect on both. Refer to *Creating and Configuring Ethernet Interfaces and Ports* in *System Interface and Port Configuration Procedures*.

With automatic card-level redundancy, there is no port-level redundancy in an MIO/UMIO failover. The standby MIO/UMIO becomes active and all ports on that card become active. The system automatically copies all the MAC addresses and configuration parameters used by the failed MIO/UMIO to its redundant counterpart. The ports on MIOs keep their original MAC addresses, and the system automatically copies the failed MIO/UMIO's configuration parameters to its redundant counterpart.

Port redundancy can be configured to be revertive or non-revertive. With revertive redundancy service is returned to the original port when service is restored.

This feature requires specific network topologies to work properly. The network must have redundant switching components or other devices that the system is connected to. The following diagrams show examples of a redundant switching topologies and how the system reacts to various external network device scenarios.

Figure 5. Network Topology Example Using MIO/UMIO Port Redundancy

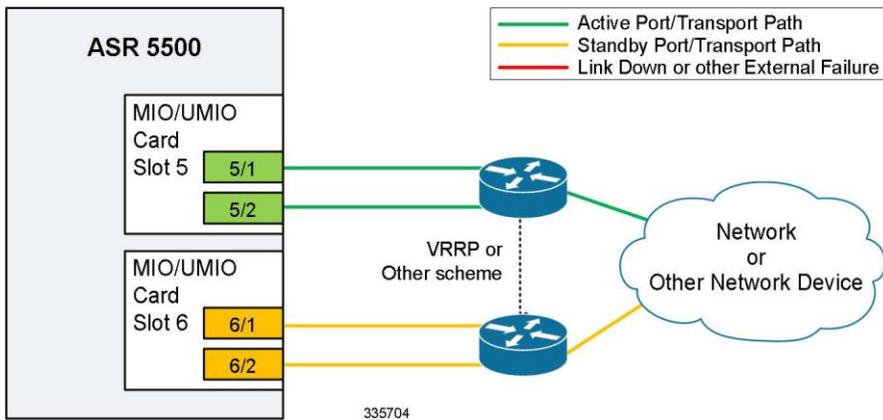
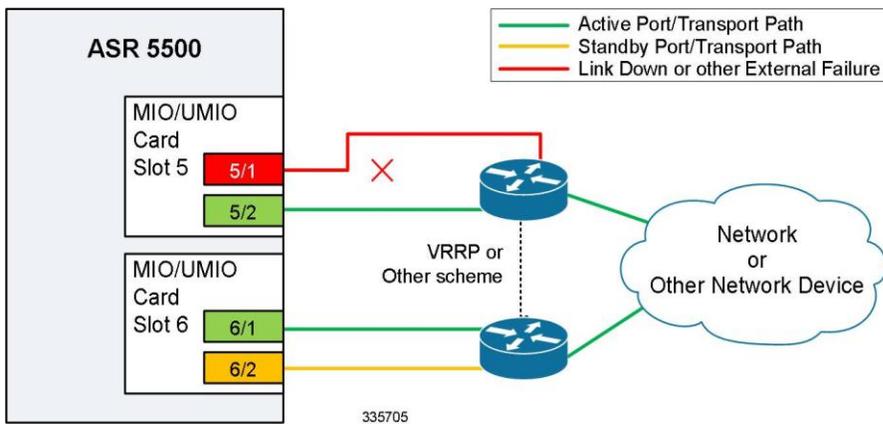
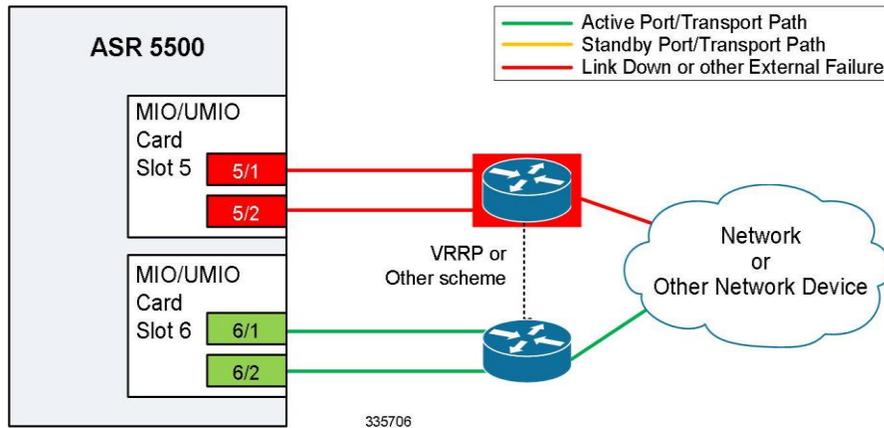


Figure 6. Port Redundancy Failover in Cable Defect Scenario



In the example above, an Ethernet cable is cut or unplugged, causing the link to go down. When this event occurs, the system, with port-mode redundancy enabled, recognizes the link down state and makes port 6/1 the active port. The switching device, using some port redundancy scheme, recognizes the failure and enables the port on the secondary switch to which the MIO/UMIO in slot 6 is connected, allowing it to redirect and transport data.

Figure 7. Port Redundancy Failover in External Network Device Failure Scenario



In the example above, a switch failure causes a link down state on all ports connected to that switch. This failure causes all redundant ports on the line card in slot 6 to move into the active state and utilize the redundant switch.

Configuring MIO/UMIO Port Redundancy Auto-Recovery

You can configure a port auto-recovery feature. When a port failure occurs and the preferred port is returned to service (link is up), control is automatically returned to that port. By default, ports are in a non-revertive state, meaning that no ports are preferred; a manual port switch is required to return use to the original port.

Important: This feature is applied on a per port basis (via the **preferred slot** keyword), allowing you to configure specific ports to be used on individual MIOs. For example, you could configure ports 10 through 19 as preferred on the MIO/UMIO in slot 5, and configure ports 20 through 29 as the preferred ports on the MIO/UMIO in slot 6.

Use the following example to configure a preferred port for revertive, automatic return to service when a problem has cleared:

```
configure
  port ethernet <slot#/port#>
    preferred slot <slot#>
  end
```

Notes

- If you do specify a preference, redundancy is revertive to the specified card. If you do not specify a preference, redundancy is non-revertive.
- Repeat for each additional port that you want to make preferred.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Verifying Port Redundancy Auto-Recovery

Verify port information by entering the following command

```
show port info slot#/port#
```

slot# is the chassis slot number of the MIO/UMIO card on which the physical port resides.

port# is the physical port on the MIO/UMIO.

The following shows a sample output of this command for port 1 on the MIO/UMIO in slot 5:

```
[local]host_name# show port info 5/1
Port: 5/1
  Port Type           : 1000 Ethernet
  Role                : Management Port
  Description         : (None Set)
  Redundancy Mode     : Port Mode
  Redundant With      : 6/1
  Preferred Port      : Non-Revertive
  Physical ifIndex    : 83951616
  Administrative State : Enabled
  Configured Duplex   : Auto
  Configured Speed    : Auto
  Configured Flow Control : Enabled
  Interface MAC Address : 02-05-47-B8-2F-41
  Fixed MAC Address   : 02-05-47-B8-2F-41
  Link State          : Up
  Link Duplex         : Full
  Link Speed          : 1000 Mb
  Flow Control        : Disabled
  Link Aggregation Group : None
  Logical ifIndex     : 83951617
  Operational State   : Up, Active
```

Configuring Data Processing Card (DPC) Availability

As discussed in the *Understanding the System Boot Process* section of *Understanding System Operation and Configuration*, when the system initially boots up, all installed DPCs and or UDPCs are placed into standby mode. You must activate some of these cards in order to configure and use them for session processing. One DPC/UDPC may remain in standby mode for redundancy.

This section describes how to activate DPCs and specify their redundancy.



Important: Refer to the *ASR 5500 Installation Guide* for information about system hardware configurations and redundancy.

Enter the following command to check the operational status of all DPCs:

```
show card table
```

This command lists the DPCs installed in the system by their slot number, their operational status, and whether or not the card is a single point of failure (SPOF).

Use the following example to configure DPC/UDPC availability:

```
configure
  card <slot_#>
    mode { active | standby }
  end
```

Notes:

- When activating cards, remember to keep at least one DPC/UDPC in standby mode for redundancy.
- Repeat for every other DPC/UDPC in the chassis that you wish to activate.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Verifying Card Configurations

Verify that the configuration was successful. Enter the following command:

```
show card table
```

Any DPC/UDPC that you made active should now have an operational status of *Active*.

Configuring ASR 5500 Link Aggregation

A Link Aggregation Group (LAG) works by exchanging control packets via Link Aggregation Control Protocol (LACP) over configured physical ports with peers to reach agreement on an aggregation of links as defined in IEEE 802.3ad. The LAG sends and receives the control packets directly on physical ports.

Link aggregation (also called trunking or bonding) provides higher total bandwidth, auto-negotiation, and recovery by combining parallel network links between devices as a single link. A large file is guaranteed to be sent over one of the links, which removes the need to address out-of-order packets.

LAG and Master Port

Logical port configurations (VLAN and binding) are defined in the master port of the LAG. If the master port is removed because of a card removal/failure, another member port becomes the master port (resulting in VPN binding change and outage), unless there is a redundant master port available.



Important: The master port on which VLAN can be created for VPN binding must always be configured on the active/master MIO/UMIO. The redundancy between MIO/UMIO 5 and MIO/UMIO 6 automatically causes both ports to be the master with the same VLANs configured and active.

LAG and Port Redundancy

ASR 5500 LAG implementation assumes that:

- LAG ports on MIO/UMIO 5 and MIO/UMIO 6 are connected to two Ethernet switches.
- LAG ports on MIO/UMIO 5 and MIO/UMIO 6 are both active at the same time.
- Ports on MIO/UMIO 5 and MIO/UMIO 6 are redundant with each other.

All ports in a LAG can be auto-switched to another MIO/UMIO when certain active port counts or bandwidth thresholds are crossed.

LAG and Multiple Switches

This feature connects subscriber traffic ports on MIOs to ports on Ethernet switches. A port failure/switch forces all ports in a LAG to switch to the other MIO/UMIO when a specified threshold is crossed. This works in a way similar to the auto-switch feature for port redundancy. LACP runs between the ASR 5500 and the Ethernet switch, exchanging relevant pieces of information, such as health status.

The following table summarizes typical LAG functionality on an MIO/UMIO card.

Table 4. MIO/UMIO LAG Functionality

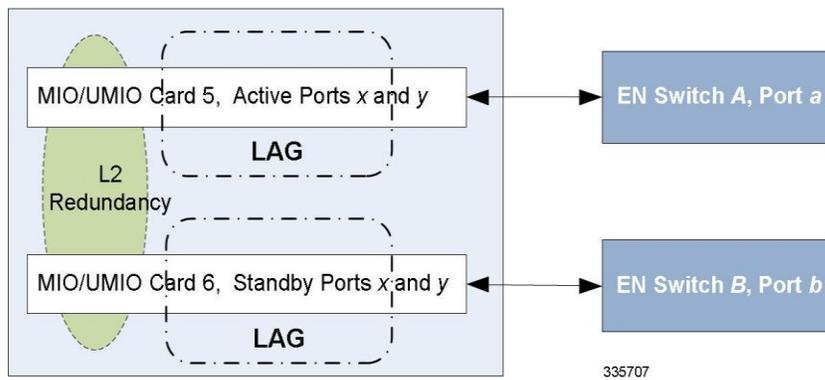
ASR 5500	LAGID	Ethernet Switch A	Ethernet Switch B
MIO/UMIO Port 11	1	Port 1	----
MIO/UMIO Port 12	1	Port 2	----
MIO/UMIO Port 13	1	----	Port 1

Multiple Switches with L2 Redundancy

To handle the implementation of LACP without requiring standby ports to pass LACP packets, two separate instances of LACP are started on redundant cards. The two LACP instances and port link state are monitored to determine whether to initiate an auto-switch (including automatic L2 port switch).

The figure below shows an LAG established across two MIO/UMIO daughter card ports with L2 redundancy.

Figure 8. LAG with L2 Redundancy, Two Ethernet Switches



An LACP implementation with L2 redundancy cannot pass traffic even though standby ports have link up. For example, with two MIO/UMIO cards connected to two different Ethernet switches and all ports in the same LAG, failure of ports would not trigger a LAG switch until the active port number ratio flipped (more ports down than up).

Port States for Auto-Switch

Ports are classified in one of four states to determine whether to start auto-switching. See the table below.

For counters, State(x) represents the number of ports on a card in that state.

Table 5. Auto-Switch Port States

State	Counter	Description
Link	L(x)	Physical link up
Standby	S(x)	Link up but in standby mode
Waiting	W(x)	Waiting for Link Aggregation Control Protocol negotiation
Aggregated	A(x)	Aggregation formed

Hold Time

Once the LAG manager switches to another LACP instance, it does not consider another change for a short period to let link and LACP negotiation settle down. This “hold time” is configurable.

The LAG manager also enters/extends the hold period when an administrator manually switches ports to trigger a card switch.

Preferred Slot

You can define which card is preferred per LAG group as a **preferred slot**. When a preferred MIO/UMIO slot is specified, it is selected for the initial timeout period to make the selection of a switch less random.

Port preference is not allowed in this mode.

Auto-Switch Criteria

The following criteria determine the switching of card x to card y to provide better bandwidth while allowing manual intervention. The evaluation of the criteria occurs outside of the hold period.

Ports are automatically switched from card x to card y when $A(y) \neq 1$, at least one port is in aggregated state on card y , and one of the following conditions is true (in order of precedence):

- $L(x) > L(y)$ Less ports with link Up on card x than card y
- $S(x) > S(y)$ More ports in Standby state on card x than card y
- $W(x) > W(y)$ More ports in Waiting state on card x than card y
- $A(x) < A(y)$ Fewer ports in Aggregated state on card x than card y
- Card y is preferred
- Card y is selected.

Link Aggregation Control

One port in an aggregation group is configured as a master so that all traffic (except control traffic) in the aggregation group logically passes through this port. It is recommended that you configure link-aggregation on the master port first when enabling LAG, and unconfigure the master port last when disabling LAG.

The following command creates link aggregation group N with port $slot\#/port\#$ as master. Only one master port is allowed for a group. N must be in the range of [1...255].

```
configure

port ethernet <slot#/port#>

link-aggregation master group <N>

exit
```

 **Important:** Link Aggregation Control Protocol (LACP) starts running only when the master port is enabled.

Use the following command to add a port as member of link aggregation group number *N* only if the master port is assigned. Otherwise, it is added to the group when the master port is assigned:

```
port ethernet <slot#/port#>

link-aggregation member group <N>

exit
```

 **Important:** The VPN can only bind the master port, and a VLAN can only be created on the master port. A failure message is generated if you attempt to bind to a link aggregation member port.

Each system that participates in link aggregation has a unique system ID that consists of a two-byte priority (where the lowest number [0] has the highest priority) and a six-byte MAC address derived from the first port's MAC address. The following command sets the system priority used to form the system ID. *P* is a hex in the range [0x0000..0xFFFF]. The default is 0x8000.

```
card <slot#>

link-aggregation system-priority <P>
```

Ports in a system are assigned keys. The group number maps directly to the key, whereupon only ports with the same key can be aggregated. Ports on each side of the link use a different aggregation key.

The system ID, port key and port ID of two peers form the Link Aggregation Group Identifier (LAGID). You can aggregate links having the same LAGID. Systems are often configured initially with each port in its own aggregation (requiring a separate key per port), or with all ports in the same aggregation (a single key for all ports). Negotiation via LACP would qualify the actual aggregation.

Systems exchange information about system ID, port key and port ID with peers across the physical links using LACP.

LACP packets are defined with the Slow Protocol format. Each system sends out its own (“actor”) information and its last received information about its peer (“partner”) over the physical link.

Use the following commands to set the LACP parameters. LACP can run in active mode to send LACP packets periodically, or in passive mode, in which it only responds to LACP packets it receives.

LACP can send packets at either a auto (30s) or fast (1s) rate. The defaults for this release are **Active** and **Auto**; see the sample configuration below:

```

config

    port ethernet <slot#/port#

        link-aggregation lACP { active | passive } [ rate { auto | fast } | timeout
        { long | short } ]

```

Peers send out LACP packets when the state changes or if a difference is found from a received LACP packet about its own state.

Corresponding ports on an MIO/UMIO redundant pair cannot be active at the same time. Redundant ports share the same MAC address, so after a failover is resolved, the original port rejoins the link aggregation group.

Redundancy Options

For L2 redundancy set the following option on the master port for use with the whole group:

```

link-aggregation redundancy standard [hold-time sec ] [preferred slot { card_number |
none } ]

```

Standard redundancy treats all cards in the group as one group.

Horizontal Link Aggregation with Two Ethernet Switches

When a LAG contains two sets of ports each connecting to a different switch, the operator has the ability to specify the slot/port (connected to the destination switch) when switching ports.

The Exec mode **link-aggregation port switch to slot/port** command configures this option. The *slot/port* is any valid port connected to the destination switch. The following criteria apply to the setting of this option:

- *slot/port* must support LAG.
- *slot/port* must be configured with LAG.
- *slot/port* must not be already actively distributing
- *slot/port* must have negotiated a link aggregation partner in standard mode.
- *slot/port*'s partner must have an equal or higher in standard mode.
- *slot/port*'s partner bundle must have equal or higher bandwidth in standard mode.
- Switching to *slot/port* must not violate preference within hold-time in standard mode.

Link Aggregation Status

To check the status of link aggregation, use the following commands:

- **show port table**
- **show port info slot/port**

A single character is used to display LAG physical port status in the output of the **show port table** command. See the table below.

Table 6. LAG Port Status

Display	Description
LA+	Port is actively used for distributing.
LA-	Port failed to negotiate LACP.
LA~ (tilde)	Port negotiated LACP but another peer was selected.
LA*	Port is (re)negotiating LACP.

Chapter 4

Management Settings

This segment provides instructions for configuring Object Request Broker Element Management (ORBEM) and Simple Network Management Protocol (SNMP) options.

This segment includes the following sections:

- [ORBEM and the Web Element Manager](#)
- [SNMP Support](#)

ORBEM and the Web Element Manager

The system can be managed by a Common Object Broker Request Architecture (CORBA)-based, element management system application called the Cisco® Web Element Manager (WEM).

You must configure the ORBEM settings on the ASR 5x00 that allow the system to communicate with the server running the WEM application.



Important: Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for detailed information about all commands.

To configure the system to communicate with the Web Element Manager:

- Step 1** Set client ID parameters and configure the STOP/TCP port settings by applying the example configuration in the [Configuring Client and Port Parameters](#) section.
- Step 2** Configure Internet Inter-ORB Protocol (IIOP) transport parameters by applying the example configuration in the [Configuring Internet Inter-ORB Protocol \(IIOP\) Transport Parameters](#) section.
- Step 3** View your new ORBEM configuration by following the steps in the [Verifying ORBEM Parameters](#) section.
- Step 4** Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring ORBEM Client and Port Parameters

Use the following example to set client ID parameters and configure the SIOP/TCP port settings:

```
configure
  orbem
    client id encrypted password <password>
    max-attempt <number>
    session-timeout <time>
    siop-port <port_number>
    event-notif-siop-port <siop_notif_port>
    event-notif-service
  end
```

Notes:

- You can issue the client id command multiple times to configure multiple clients.
- If a client ID is de-activated due to reaching the configured maximum number of attempts, use the **activate client id** command to reactivate it.

- If a firewall exists between the system and the Web Element Manager, open the SIOP port number and the TCP port number 15011.
- If the ORB Notification Service is enabled via the **event-notif-service** command, you can set filters to determine which events are to be sent. By default, the Service sends all error and higher level events, “info” level events for the ORBS facility, CLI command logs, and license change logs. Optionally, configure a filter by including the **event-notif-service filter** command. Enter this command for each filter you need to configure.

Configuring IIOP Transport Parameters

Use the following example to configure Internet Inter-ORB Protocol (IIOP) transport parameters that enable ORB-based management to be performed over the network:

```
configure
orbem
  iiop-transport
  iiop-port <iiop_port_number>
  event-notif-iiop-port <iiop_notif_port>
end
```

Notes:

- If you are using the Secure Sockets Layer (SSL) option, do not enable the IIOP transport parameter. The WEM default process enforces SSL.
- You configure the ORBEM interface to use SSL by specifying a certificate and private key.

Verifying ORBEM Parameters

- Step 1** Run the **show orbem client table** command to verify that the client was configured properly. This command lists the configured ORBEM clients and displays their state and privileges.
- Step 2** Run the **show orbem status** command to verify the ORBEM parameter configuration. The following displays a sample of this command's output.

```
Service State                : On
Management Functions        : FCAPS
IOP Address                  : 192.168.1.150
SSL Port                     : 14131
TCP Port                     : 14132
Notification SSL Port       : 7777
Notification TCP Port       : 7778
Session Timeout              : 86400 secs
Max Login Attempts          : 5
IIOP Transport               : On
Notification                  : On
Debug Level                  : Off
IDL Version Check           : On
Number of Current Sessions   : 1
Number of Event Channels Open : 0
Number of Operations Completed : 2895
Number of Events Processed   : 0
Avg Operation Processing time : 87214 usecs
                             (last 1000) : 87950 usecs
```

SNMP Support

The system uses the SNMP to send traps or events to the Web Element Manager server or an alarm server on the network. You must configure SNMP settings to communicate with those devices.

 **Important:** Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

The *SNMP MIB Reference* describes the MIBs and SNMP traps supported by the ASR 5x00 platform.

To configure the system to communicate with the WEM server or an alarm server:

- Step 1** Set SNMP parameters such as UDP port, and alarm server target by applying the example configuration in the [Configuring SNMP and Alarm Server Parameters](#) section.
- Step 2** To view your new SNMP configuration, follow the steps in the [Verifying SNMP Parameters](#) section.
- Step 3** Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring SNMP and Alarm Server Parameters

Use the following example to set SNMP and alarm server parameters:

```
configure

  system contact <contact_name>

  system location <location_name>

  snmp authentication-failure-trap

  snmp community <community_string>

  snmp server port <port_number>

  snmp target <name ip_address>

  snmp engine-id local <id_string>

  snmp notif-threshold <value> low <low_value> period <time_period>

  snmp user <user_name>

  snmp mib <mib_name>

end
```

Notes:

- The **system contact** is the name of the person to contact when traps are generated that indicate an error condition.
- An **snmp community** string is a password that allows access to system management information bases (MIBs).
- The system can send SNMPv1, SNMPv2c, or SNMPv3 traps to numerous target devices. However, WEM can only process SNMP version 1 (SNMPv1) and SNMP version 2c (SNMPv2c) traps. If the SNMP target you are configuring is the WEM application, use the **snmp target** command to configure use of version 1 or version 2c. Issue this command as many times as you need to configure multiple targets. If you configure multiple targets, generated alarms are sent to every configured target.
- The **snmp notif-threshold** command configures the number of SNMP notifications that need to be generated for a given event and the number of seconds in the monitoring window size (default = 300), before the notification is propagated to the SNMP users (default = 100).
- The **snmp engine-id local** command is optional. It is only required if your network requires SNMP v3 support. The engine ID uniquely identifies the SNMP engine and associated SNMP entities, thus providing a security association between the two for the sending and receiving of data.
- The **snmp user** name is for SNMP v3 and is optional. There are numerous keyword options associated with this command.
- Use the **snmp mib** command to enable other industry standard and Cisco MIBs. By default only the STARENT-MIB is enabled.



Important: SNMPv3 traps are not supported by the WEM application.

Verifying SNMP Parameters

Step 1 Run the **show snmp server** command to verify that the SNMP server information is correctly configured. The following displays a sample output of this command.

```
SNMP Server Configuration:
  Server State           : enabled
  SNMP Port              : 161
  sysLocation           : chicago
  sysContact             : admin
  authenticationFail traps : Enabled
  EngineID              : 123456789
  Alert Threshold        : 100 alerts in 300 seconds
  Alert Low Threshold    : 20 alerts in 300 seconds
SNMP Agent Mib Configuration:
  STARENT-MIB           : Enabled
  IF-MIB                : Disabled
  ENTITY-MIB            : Disabled
  ENTITY-STATE-MIB      : Disabled
  ENTITY-SENSORE-MIB    : Disabled
  HOST-RESOURCES-MIB    : Disabled
  CISCO-MOBILE-WIRELESS-SERVICE-MIB : Disabled
  CISCO-ENTITY-DISPLAY-MIB : Disabled
  CISCO-PROCESS-MIB     : Disabled
  CISCO-ENTITY-FRU-CONTROL-MIB : Disabled
```

Step 2 Verify that the SNMP community(ies) were configured properly by entering the following command:

```
show snmp communities
```

The output of this command lists the configured SNMP communities and their corresponding access levels.

Step 3 Verify that the SNMP transports are configured properly by entering the following command:

```
show snmp transports
```

The following displays a sample output:

```
Target Name:    rms1
IP Address:    192.168.1.200
Port:         162
Default:      Default
Security Name: public
Version:      1
Security:
View:
Notif Type:   traps
```

Controlling SNMP Trap Generation

The system uses SNMP traps (notifications) to indicate that certain events have occurred. By default, the system enables the generation of all traps. However, you can disable individual traps to allow only traps of a certain type or alarm level to be generated. This section provides instructions for disabling/enabling SNMP traps.

 **Important:** Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To configure SNMP trap generation:

Step 1 Set parameters by applying the following example configuration:

```
configure

snmp trap suppress

snmp trap suppress <trap_name1> <trap_name2> ... <trap_nameN>
```

 **Important:** If at a later time you wish to re-enable a trap that was previously suppressed, use the **snmp trap enable** command.

```
snmp trap enable <trap_name1> <trap_name2> ... <trap_nameN> target <target-name>
```

Step 2 Save the configuration as described in *Verifying and Saving Your Configuration*.

Chapter 5

Verifying and Saving Your Configuration

This segment describes how to save your system configuration and includes the following sections:

- [Verifying the Configuration](#)
- [Saving the Configuration](#)

Verifying the Configuration

You can use a number of commands to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of or specific lines in the configuration file.

Feature Configuration

In many configurations, you have to set and verify specific features. An example includes IP address pool configuration. Using this example, enter the following commands to verify proper feature configuration:

Enter the **show ip pool** command to display the IP address pool configuration. The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:
+-----Type:      (P) - Public      (R) - Private
|                 (S) - Static      (E) - Resource
|
|+-----State:    (G) - Good        (D) - Pending Delete  (R)-Resizing
||
||+---Priority: 0..10 (Highest (0) .. Lowest (10))
||||
||||+---Busyout: (B) - Busyout configured
||||||
vvvvvv Pool Name                Start Address   Mask/End Address Used   Avail
-----
PG00 ipsec                      12.12.12.0     255.255.255.0  0     254
PG00 pool1                      10.10.0.0      255.255.0.0   0     65534
SG00 vpnpool                    192.168.1.250  192.168.1.254  0     5

Total
Pool Count: 5
```

 **Important:** To configure features on the system, use the **show** commands specifically for these features. Refer to *Exec Mode show Commands* in the *Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

```
show service_type service_name
```

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called *pgw* is configured.

```

Service name           : pgw1
Service-Id            : 1
Context               : test1
Status                : STARTED
Restart Counter       : 8
EGTP Service          : egtpl
LMA Service           : Not defined
Session-Delete-Delay Timer : Enabled
Session-Delete-Delay timeout : 10000 (msecs)
PLMN ID List          : MCC: 100, MNC: 99
Newcall Policy        : None

```

Context Configuration

Verify that your context was created and configured properly by entering the **show context name** *name* command.

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```

Context Name      ContextID      State
-----
test1            2              Active

```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the **show configuration** command.

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the **show configuration errors** command.

This command displays errors it finds within the configuration. For example, if you have created a service named “servicel”, but entered it as “srvl” in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu as shown in the examples below.

```
show configuration errors section ggsn-service
```

or

```
show configuration errors section aaa-config
```

If the configuration contains no errors, an output similar to the following is displayed:

```

#####
Displaying Global
AAA-configuration errors
#####
Total 0 error(s) in this section !

```

Saving the Configuration

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

```
save configuration url [-redundant] [-noconfirm] [obsolete-encryption]
[showsecrets] [verbose]
```

Table 7. Command Syntax for Saving the Configuration

Keyword/Variable	Description
<i>url</i>	Specifies the path and name to which the configuration file is to be stored. <i>url</i> may refer to a local or a remote file. <i>url</i> must be entered using one of the formats listed below.
	file: { /flash usb1 /hd-raid/ } [/directory]/<filename> }
	tftp:// { <i>ipaddress</i> <i>host_name</i> [: <i>port#</i>] } [/ <i>directory</i>] / <i>filename</i>
	ftp:// [<i>username</i> [: <i>pwd</i>] @] { <i>ipaddress</i> <i>host_name</i> } [: <i>port#</i>] [/ <i>directory</i>] / <i>filename</i>
	sftp:// [<i>username</i> [: <i>pwd</i>] @] { <i>ipaddress</i> <i>host_name</i> } [: <i>port#</i>] [/ <i>directory</i>] / <i>filename</i>
	/flash corresponds to the flash memory on the active MIO/UMIO.
	/usb1 corresponds to the USB port on the active MIO/UMIO.
	/hd-raid corresponds to the RAID array of solid state drives supported on the FSCs and accessed via the MIO/UMIO.
	<i>ipaddress</i> is the IP address of the network server. <i>host_name</i> is the network server's <i>hostname</i> . <i>port#</i> is the network server's logical port number. Defaults are:
	<ul style="list-style-type: none"> • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data
<p>NOTE: <i>host_name</i> can only be used if the networkconfig parameter is configured for DHCP and the DHCP server returns a valid <i>nameserver</i>.</p> <p><i>username</i> is the username required to gain access to the server if necessary.</p> <p><i>password</i> is the password for the specified username if required.</p> <p><i>/directory</i> specifies the directory where the file is located if one exists.</p> <p><i>/filename</i> specifies the name of the configuration file to be saved.</p> <p>NOTE: Configuration files should be named with a .cfg extension.</p>	

Keyword/Variable	Description
-redundant	<p><i>Optional:</i> This keyword directs the system to save the CLI configuration file to the local device defined by the <i>url</i> variable, and then automatically copy that same file to the like device on the standby MIO/UMIO, if available.</p> <p>NOTE: This keyword will only work for similar local devices that are located on both the active and standby MIOs. For example, if you save the file to the /usb1 device on the active MIO/UMIO, that same type of device (a memory stick in the USB port of the standby MIO/UMIO) must be available. Otherwise, a failure message is displayed.</p> <p>NOTE: This keyword is only applicable when saving a configuration file to local devices.</p> <p>NOTE: When saving the file to an external network (non-local) device, the system disregards this keyword.</p>
-noconfirm	<i>Optional:</i> Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).
obsolete-encryption	Saves configuration with encrypted values generated from an obsolete encryption method.
showsecrets	<i>Optional:</i> Saves the CLI configuration file with all passwords in plain text, rather than their default encrypted format.
verbose	<i>Optional:</i> Displays every parameter that is being saved to the new configuration file.



Important: Do not use the “/” (forward slash), “:” (colon) or “@” (at sign) characters when entering a string for the following URL fields: directory, filename, username, password, host or port#.

To save a configuration file called *system.cfg* to a directory that was previously created called *cfgfiles* to the flash memory on the active MIO/UMIO, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called *simple_ip.cfg* to a directory called *host_name_configs*, using an FTP server with an IP address of *192.168.34.156*, on which you have an account with a username of *administrator* and a password of *secure*, use the following command:

```
save configuration
sftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called *init_config.cfg* to the *root* directory of a TFTP server with a hostname of *config_server*, enter the following command:

```
save configuration tftp://config_server/init_config.cfg
```


Chapter 6

System Interfaces and Ports

This segment describes how to create a context and configure system interfaces and ports within the context. Before beginning these procedures, refer to your product-specific administration guide for configuration information for your product.

This segment includes the following:

- [Creating Contexts](#)
- [Creating and Configuring Ethernet Interfaces and Ports](#)

 **Important:** Make sure at least one Data Processing Card (DPC) or Data Processing Card Universal (UDPC) is active before you configure system interfaces and ports. Refer to *System Settings* in this guide for information and instructions on activating DPCs.

Creating Contexts

Even though multiple contexts can be configured to perform specific functions, they are all created using the same procedure.



Important: Commands used in the configuration examples in this section represent the most common or likely commands and/or keyword options. In many cases, other commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To create a context, apply the following example configuration:

```
configure
  context <name>
  end
```

Repeat to configure additional contexts.

Viewing and Verifying Contexts

Step 1 Verify that your contexts were successfully created by entering the following command:

```
show context all
```

The output is a two-column table similar to the example below. This example shows that two contexts were created: one named *source* and one named *destination*.

Context Name	ContextID	State
-----	-----	----
local	1	Active
source	2	Active
destination	3	Active

The left column lists the contexts that are currently configured. The center column lists the corresponding context ID for each of the configured contexts. The third column lists the current state of the context.

Step 2 Save your configuration as described in *Verifying and Saving Your Configuration*.

Step 3 Now that the context has been created, interfaces and specific functionality can be configured within the context. Proceed to other sections for instructions on configuring specific services and options.

Creating and Configuring Ethernet Interfaces and Ports

Regardless of the type of application interface, the procedure to create and configure it consists of the following:

- Step 1** Create an interface and assign an IP address and subnet mask to it by applying the example configuration in the [Creating an Interface](#) section.
- Step 2** Assign a physical port for use by the interface and bind the port to the interface by applying the example configuration in the [Configuring a Port and Binding it to an Interface](#) section.
- Step 3** Optionally configure a static route for the interface by applying the example configuration in the [Configuring a Static Route for an Interface](#) section.
- Step 4** Repeat the above steps for each interface to be configured.



Important: This section provides the minimum instructions for configuring interfaces and ports to allow the system to communicate on the network. Commands that configure additional interface or port properties are described in *Ethernet Port Configuration Mode Commands* and *Ethernet Interface Configuration Mode Commands* in the *Command Line Interface Reference*.



Caution: To ensure that system line card and port-level redundancy mechanisms function properly, the Spanning Tree protocol must be disabled on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

Creating an Interface

Use the following example to create a new interface in a context:

```
configure
  context <name>
    interface <name>
      { ip | ipv6 } address <address subnetmask> [ secondary ]
    end
```

Notes:

- *Optional:* Add the **loopback** keyword option to the **interface** *name* command, to set the interface type as “loopback” which is always UP and not bound to any physical port.
- *Optional:* Add the **secondary** keyword to the { **ip** | **ipv6** } **address** command, to assign multiple IP addresses to the interface. IP addresses can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- *Optional:* In the interface config mode, add the **port-switch-on-L3-fail address** command, to configure the interface for switchover to the port on the redundant line card if connectivity to a specified IP address is lost. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

Configuring a Port and Binding It to an Interface

Use the following example configuration to configure and assign a port to an interface:

```
configure
    port ethernet <slot#/port#>
        description <description>
        no shutdown
        bind interface <interface_name> <context_name>
    end
```

Notes:

- For **port ethernet slot#**, use the actual chassis slot in which the MIO/UMIO card is installed; this could be 5 or 6.
- For **port ethernet port#**, use ports 10 to 19 (DC1) or 20 to 29 (DC2).
- *Optional:* In the Ethernet Port configuration mode, add the preferred **slot slot#** command if you want to specify a port preference.
- Binding associates the port and all of its settings to the named interface.

Configuring a Static Route for an Interface

Use the following example to configure a static route for an interface:

```
configure
    context <name>
        { ip | ipv6 } route <ip_address> <netmask> next-hop <gw_address>
        <interface_name>
    end
```

Notes:

- *ip_address* and *netmask* are the IP address and subnet mask of the target network. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- *gw_address* is the IP address of the default gateway or next-hop route. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- To configure a route to the gateway router, use 0.0.0.0 for the network and mask variables.
- Repeat as needed. Multiple static routes can be configured to the same destination to provide an alternative means of communication in case the preferred route fails.

Viewing and Verifying Port Configuration

Step 1 Verify that your interface configuration settings are correct by entering the following commands:

```
context context_name

show { ip | ipv6 } interface
```

context_name represents the name of the context in which the interface was created. The output from these commands should be similar to the following example.

In this example an interface named *mgmt1* was configured in the local context.

```
Intf Name:      mgmt1
Intf Type:      Broadcast
IP State:       UP (Bound to 5/11 untagged, ifIndex 285278209)
IP Address:     192.168.100.3      Subnet Mask: 255.255.255.0
Bcast Address:  192.168.100.255   MTU:          1500
Resoln Type:    ARP              ARP timeout:   3600 secs
Number of Secondary Addresses:  0
Total interface count:          1
```

Step 2 Verify that your port configuration settings are correct by entering the following command:

```
show configuration port slot#/port#
```

slot# is the chassis slot number of the MIO/UMIO on which the physical port resides. *slot#* can be 5 or 6.

This command produces an output similar to that displayed in the following example that shows the configuration for port 11 on the MIO/UMIO installed in chassis slot 5.

In this example, the port is bound to an interface called *rp1* configured in a context called *source*.

```
config
port ethernet 5/11
description MIO5/11_RP1
no shutdown
bind interface rp1 source
#end
```

Step 3 Verify that your static route(s) was configured properly by entering the following command:

```
show ip static-route
```

This command produces an output similar to that displayed in the following example that shows a static route to a gateway with an IP address of 192.168.250.1.

Destination	Nexthop	Protocol	Prec	Cost	Interface
0.0.0.0/0	192.168.250.1	Static	0	0	MIO1
0.0.0.0/0	192.168.250.1	Static	0	0	rp1 source

Step 4 Save the configuration as described in *Verifying and Saving Your Configuration*.

Chapter 7

System Security

This segment describes the security features supported on the ASR 5500 platform.

This segment explores the following topics:

- [Per-Chassis Key Identifier](#)
- [Encrypted SNMP Community Strings](#)
- [Lawful Intercept Restrictions](#)
- [Adding, Modifying and Removing Users](#)
- [Hidden Commands](#)

Per-Chassis Key Identifier

A user can set a unique chassis key which will work only for a chassis or for any set of chassis that will share the same configuration information.

The chassis key consists of 1 to 16 alphanumeric ASCII characters. The chassis key plain-text value is never displayed to the user; it is entered interactively and not echoed to the user.

On the ASR5500 the encrypted chassis key is stored in the midplane EEPROM and shared by both MIO/UMIOs.

If the chassis key identifier stored in the header comment line of the configuration file does not match the chassis key, an error message is displayed to the user. The user can change the chassis key value simply by entering the chassis key again. The previous chassis key is replaced by a new chassis key. The user is not required to enter a chassis key.

If the user does not configure a chassis key, the system generates a unique value for that chassis.

 **Important:** Changing a chassis key may invalidate previously generated configurations. This is because any secret portions of the earlier generated configuration will have used a different encryption key. For this reason the configuration needs to be recreated and restored.

 **Important:** To make password configuration easier for administrators, the chassis key should be set during the initial chassis set-up.

The configuration file contains a one-way encrypted value of the chassis key (the chassis key identifier) and the version number in a comment header line. These two pieces of data determine if the encrypted passwords stored within the configuration will be properly decrypted.

While a configuration file is being loaded, the chassis key used to generate the configuration is compared with the stored chassis key. If they do not match the configuration is not loaded.

The user can remove the chassis key identifier value and the version number header from the configuration file. Also, the user may elect to create a configuration file manually. In both of these cases, the system will assume that the same chassis key will be used to encrypt the encrypted passwords. If this is not the case, the passwords will not be decrypted due to resulting non-printable characters or memory size checks. This situation is only recoverable by setting the chassis key back to the previous value, editing the configuration to have the encrypted values which match the current chassis key, or by moving the configuration header line lower in the configuration file.

MIO/UMIO Synchronization

On boot up both MIO/UMIOs automatically read the chassis key configured on the ASR 5500 midplane.

Protection of Passwords

Users with privilege levels of Inspector and Operator cannot display decrypted passwords in the configuration file via the ASR 5500 command line interface (CLI).

Secure Configuration Password Encryption

The system encrypts passwords using an MD5-based cipher. These passwords also have a random 64-bit (8-byte) salt added to the password. The chassis key is used as the encryption key.

Using the chassis key allows for an encryption method where the decryption requires the knowledge of a “shared secret”. Only a chassis with knowledge of this shared secret can access the passwords. To decipher passwords, a hacker who knew the chassis key would still need to identify the location of the 64-bit random salt value within the encryption.

The encrypted password is displayed with a prefixed of “+A” in the configuration file to identify the methodology used for encrypting.

Support for Non-Current Encryptions and Decryptions

The system supports previously formatted encrypted passwords. The syntax of the encrypted passwords indicates to the ASR 5500 which methodology was used for encryption. If the system does not see a prefix before the encrypted password, the earlier encryption method using a fixed key will be used. If the encrypted password includes the “+A” prefix, the decryption method uses the chassis key and random salt.

If the user saves a new configuration, the generated file will always contain passwords encrypted by the most recent method. The user cannot generate the earlier DES-based encryption values. However, all future StarOS releases will continue to support plain-text password entry for all two-way encryptable passwords

The recommended process for changing the chassis key without causing a “lock-out” state is as follows:

- Load the configuration file of the last good configuration using the previous chassis key.
- Change the chassis key to the new desired value.
- Save the configuration with this new chassis key.

Refer to *Configuring a Chassis Key* in *System Settings* for additional information.

Support for ICSR Configurations

Inter-Chassis Session Recovery (ICSR) is a redundancy configuration that employs two identically configured ASR 5500 chassis as a redundant pair.

ICSR chassis share the same chassis key. If the ICSR detects that the two chassis have incompatible chassis keys, an error message is logged but the ICSR system will continue to run. Without the matching chassis key, the standby ICSR chassis can recover services if the active chassis goes out of service; the standby chassis will still have access to the passwords in their decrypted form.

ICSR chassis use Service Redundancy Protocol (SRP) to periodically check to see if the redundancy configuration matches with either decrypted passwords or DES-based two-way encryption strings. Since the configuration is generated internally to the software, users are not able to access the configuration used to check ICSR compatibility.

Encrypted SNMP Community Strings

Simple Network Management Protocol (SNMP) uses community strings as passwords for network elements. Although these community strings are sent in clear-text in the SNMP PDUs, the values can be encrypted in the configuration file.

The **snmp community encrypted name** command enables the encryption of SNMP community strings. For additional information, see *Global Configuration Mode Commands* in the *CLI Reference*.

Lawful Intercept Restrictions

This section describes some of the security features associated with the provisioning of Lawful Intercept (LI). For additional information, refer to the *Lawful Intercept Configuration Guide*.

LI Server Addresses

An external authenticating agent (such as RADIUS or Diameter) sends a list of LI server addresses as part of access-accept. For any intercept that was already installed or will be installed for that subscriber, a security check is performed to match the LI server address with any of the LI-addresses that were received from the authenticating agent. Only those addresses that pass this criteria will get the intercepted information for that subscriber.

While configuring a campon trigger, the user will not be required to enter the destination LI server addresses. When a matching call for that campon trigger is detected, a security check is done with the list received from the authentication agent. The LI-related information is only forwarded if a matching address is found.

When an active-only intercept is configured, if a matching call is found, a security check is made for the LI address received from the authentication agent and the intercept configuration will be rejected.

If no information related to LI server addresses is received for that subscriber, LI server addresses will not be restricted.



Important: A maximum of five LI server addresses are supported via an authenticating agent.

Modifying Intercepts

One LI administrator can access and/or modify the intercepts created by another LI administrator. Whenever an intercept is added, removed or modified, an event log is displayed across LI administrators about the change. An SNMP trap is also generated.

Adding, Modifying and Removing Users

It is considered uncommon for a user to be added or removed from the ASR 5500. Likewise, it is considered uncommon for a user's privileges to be modified. However, if the system is compromised, it is common for attackers to add or remove a privileged user, raise their privileges or lower the privileges of others.

As a general rule, lower privileged users should not be allowed to increase their privileges or gain access to sensitive data, such as passwords, which were entered by higher privileged users.

 **Important:** The ASR 5500 can only detect changes in users and user attributes, such as privilege level, when these users are configured through the ASR 5500.

Notification of Users Being Added or Deleted

Users with low level authorization should not be able to create users with high level authorization. However, if a malicious actor were to be able to create a high level authorized user, they could then delete the other high level authorized users, thereby locking them out of the system.

The following SNMP traps notify an administrator when users are added or removed:

- **starLocalUserAdded** – indicates that a new local user account has been added to the system.
- **starLocalUserRemoved** – indicates that a local user account has been removed from the system.

Notification of Changes in Privilege Levels

Whenever a user's privilege level is increased or decreased, an SNMP notification will be sent out. A malicious actor may gain access to more privileged commands by somehow promoting their privileges. Once this is done, they could then “demote” the privileges of all the other users, thereby locking the proper administrators out of the system.

The **starLocalUserPrivilegeChanged** trap indicates that a local user's privilege level has been changed.

User Access to Operating System Shell

The **starOsShellAccessed** trap indicates that a user has accessed the operating system shell.

Hidden Commands

Users with Security Administrator privilege can enable the display of previously hidden commands. The CLI test-commands mode displays new command keywords for existing commands, as well as new commands.

 **Caution:** CLI test-commands are intended for diagnostic use only. Access to these commands is not required during normal system operation. These commands are intended for use by Cisco TAC personnel only. Some of these commands can slow system performance, drop subscribers, and/or render the system inoperable.

Enabling cli test-commands Mode

To display hidden commands, the user must log into the CLI as a Security Administrator and go to the Global Configuration mode.

Enter **cli hidden** to enable the use of hidden commands.

This command sequence is shown below.

```
[local] asr5x00# config
[local] asr5x00(config)# cli hidden
[local] asr5x00(config)#
```

 **Important:** Low-level diagnostic and test commands/keywords will now be visible to a user with Administrator or higher privilege. There is no visual indication on the CLI that the test-commands mode has been enabled.

Enabling Password for Access to CLI-test commands

A Security Administrator can set a plain-text or encrypted password for access to CLI test commands. The password value is stored in **/flash** along with the boot configuration information. The **show configuration** and **save configuration** commands will never output this value.

The Global Configuration mode command **tech-support test-commands [encrypted] password <password>** sets an encrypted or plain-text password for access to CLI test-commands.

This command sequence is shown below.

```
[local] asr5x00# config
[local] asr5x00(config)# tech-support test-commands password <password>
[local] asr5x00(config)#
```

When a test-commands password is enabled, the Global Configuration mode command **cli test-commands [encrypted] password <password>** requires the entry of the password keyword. If the **encrypted** keyword is specified, the **<password>** argument is interpreted as an encrypted string containing the password value. If the **encrypted** keyword is not specified, the **<password>** argument is interpreted as the actual plain text value

 **Important:** If **tech-support test-commands password** is never configured, **cli-test commands** will always fail. If the **password** keyword is not entered for **cli test-commands**, the user is prompted (no-echo) to enter the password. Also, **cli hidden** must be enabled by a Security Administrator to access the CLI test-commands.

Exec Mode cli test-commands

Exec mode commands are available to a privileged user who enters the command **cli test-commands** from Exec mode.

```
[local]asr5x00# cli test-commands password <password>  
Warning: Test commands enables internal testing and debugging commands  
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```



Important: An SNMP trap (**starTestModeEntered**) is generated whenever a user enters CLI test-commands mode.

Configuration mode cli test-commands

Configuration commands which provided access to low-level software parameters are accessible only after a privileged user enters the command **cli test-commands** from Global Configuration mode.

```
[local]asr5x00# config  
[local]asr5x00(config)# cli test-commands password <password>  
Warning: Test commands enables internal testing and debugging commands  
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```



Important: An SNMP trap (**starTestModeEntered**) is generated whenever a user enters CLI test-commands mode.

Chapter 8

Software Management Operations

This segment provides information about software management operations on the system. Software management sections include:

- [Understanding the Local File System](#)
- [Maintaining the Local File System](#)
- [Configuring the Boot Stack](#)
- [Upgrading the Operating System Software](#)
- [Performing Dynamic Software Updates](#)
- [Managing License Keys](#)
- [Managing Local-User Administrative Accounts](#)

Understanding the Local File System

The Management Input/Output (MIO) or Management Input/Output Universal (UMIO) card provides control and management for the system.

The local file system is made up of files that are stored on one or more of the following:

- **/flash** - Flash memory located on the circuit board of the MIO/UMIO, is the default storage media for the operating system software image, CLI configuration, and crash log files used by the system.
- **/usb1** - This device is available when a USB memory stick is inserted on the front panel of the active MIO/UMIO.
- **/hd-raid** - This is the solid state hard disk array supported by the Fabric and Storage Cards (FSCs) and accessed via the active MIO/UMIO.

File Types Used by the Local File System

The following file types can be located in the local file system:

- **Operating System Software Image File:** This binary file type is identified by its **.bin** extension. The file is the operating system that is loaded by the system upon startup or reloading. This is an executable, read-only file that cannot be modified by end users.
- **CLI Configuration File:** This file type is identified by its **.cfg** extension. These are text files that contain CLI commands that work in conjunction with the operating system software image. These files determine services to be provided, hardware and software configurations, and other functions performed by the system. The files are typically created by the end user. You can modify the files both on and off-line and use descriptive long filenames.
- **System File:** Only one file identified by a **.sys** extension is used by the system. The **boot.sys** file contains system-specific information, which describes how the system locates, and in what priority it loads, file groups (paired **.bin** and **.cfg** files) from its boot stack.
- **Abridged Crash Log:** The abridged crash log, identified by its **crashlog** filename, contains summary information about software or hardware failures that occur on the system. This file is located in the **/flash/crsh2/** directory on the device. You can view the contents of this file through the CLI, but you cannot modify the file.

Understanding the boot.sys File

The system uses the **boot.sys** file to store the prioritized boot stack parameters and file groups the system uses during startup. Modify this file only through system CLI commands and not through external means. Boot parameters contain information the system needs to locate the operating system image file, including:

- **bootmode:** This setting is typically configured to normal, and identifies how the system starts.
- **network interface configuration:** Use these optional boot method settings when you configure the system to obtain its operating system image from an external network server that is using one of the management LAN interfaces on the MIO/UMIO card.
- **boot stack information:** The boot stack is made up of prioritized file group entries that designate the operating system image file and the CLI configuration file to load.

When a system is unpacked and started for the first time, the `boot.sys` file is configured to use the normal boot mode and load the operating system software image from the `/flash` directory.

There is no CLI configuration file contained on the local file system. This causes the system to automatically start its CLI-based Quick Setup Wizard upon the first successful boot. Refer to *Getting Started* for more information on using the Quick Setup Wizard.

Maintaining the Local File System

Use CLI commands to manage and maintain the devices that make up the local file system. Execute all the commands described in this section in the Exec Mode. Unless otherwise specified, you must have security administrator or administrator privileges to execute these commands.

File System Management Commands

Use the commands in this section to manage and organize the local file system.

Synchronizing the File System

Commands are supported for mirroring the local file systems from the active MIO/UMIO to the standby MIO/UMIO in systems containing two cards. Use these commands to synchronize any or all of the local devices.

 **Important:** Crash log files are not synchronized when these commands are executed.

The following command synchronizes the file systems between two MIO/UMIOs:

```
filesystem synchronize { / flash | /usb1 | a ll } [checkonly] [from | to ]
```

Table 8. Command Syntax Descriptions

Keyword/Variable	Description
<code>/flash</code>	Synchronizes only the file system on the standby MIO/UMIO.
<code>/usb1</code>	Synchronizes only the file system of the USB stick installed in the standby MIO/UMIO.
<code>all</code>	Specifies that filesystems on all available matching local devices be synchronized. Notes: Only filesystems on matching local devices are synchronized. For example, if the active MIO/UMIO contains two local devices (<code>/flash</code> and <code>/usb1</code>) and the standby MIO/UMIO contains only one local device (<code>/flash</code>), then synchronization only occurs on the matching local device (<code>/flash</code>).
<code>checkonly</code>	Displays a list of files that can be synchronized without executing any synchronization actions.
<code>from slot# to slot#</code>	Specifies the direction of the file copying by slot number.

The following command synchronizes the file systems on two MIO/UMIO `/flash` devices.

```
filesystem synchronize /flash
```

Creating Directories

Use the **mkdir** command to create a new directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

```
mkdir { /flash | /usb1 | /hd-raid } /dir_name
```

Use the following command to create a directory named *configs*:

```
mkdir /flash/configs
```

Renaming Files and Directories

Use the **rename** command to change the name of a file from its original name to a different name. Remember to use the same file extension, if applicable, to ensure that the file type remains unchanged.

```
rename { /flash | /usb1 | /hd-raid } /src_filename { /flash | /usb1 | /hd-raid }  
/dst_filename [-noconfirm]
```

Table 9. rename Command Options

Keyword/Variable	Description
<i>src_filename</i>	The name of the source file, with its extension, that you are renaming.
<i>dst_filename</i>	The name of the destination file, with its extension, to which the source file is being renamed. Be sure to use the same file extension to ensure that the file type remains unchanged.
-noconfirm	Executes the command without displaying the “Are you sure? [Yes No]” confirmation prompt.

Use the following command to rename a file named *iot_test.cfg* to *iot_accept.cfg* on the */flash* local device.

```
rename /flash/iot_test.cfg /flash/iot_accept.cfg -noconfirm
```

 **Important:** Use the **rename** command only within the same local device. You cannot rename a file and place it onto another local device at the same time. To move a renamed file, you must use the **copy** command.

Copying Files on the ASR 5500 Chassis

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

```
copy from_url to_url [-noconfirm]
```

Table 10. Command Syntax for Saving the Configuration

Keyword/Variable	Description
<i>from_url</i> <i>to_url</i>	Specifies the path and name from which the configuration file is to be copied and the one to which it will be written. <i>url</i> may refer to a local or a remote file. <i>url</i> must be entered using one of the formats listed below.
	file: { /flash usb1 /hd-raid [<i>/directory</i>] /< <i>filename</i> > }
	tftp:// { <i>ipaddress</i> <i>host_name</i> [<i>:port#</i>] } [<i>/directory</i>] / <i>filename</i>
	ftp:// [<i>username</i> [<i>:pwd</i>] @] { <i>ipaddress</i> <i>host_name</i> } [<i>:port#</i>] [<i>/directory</i>] / <i>filename</i>
	sftp:// [<i>username</i> [<i>:pwd</i>] @] { <i>ipaddress</i> <i>host_name</i> } [<i>:port#</i>] [<i>/directory</i>] / <i>filename</i>
	/flash corresponds to the flash memory on the active MIO/UMIO.
	/usb1 corresponds to the USB port on the active MIO/UMIO.
	/hd-raid corresponds to the RAID array of solid state drives supported on the FSCs and accessed via the MIO/UMIO.
	<i>ipaddress</i> is the IP address of the network server. <i>host_name</i> is the network server's <i>hostname</i> . <i>port#</i> is the network server's logical port number. Defaults are: <ul style="list-style-type: none"> • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data
	NOTE: <i>host_name</i> can only be used if the networkconfig parameter is configured for DHCP and the DHCP server returns a valid nameserver.dx <i>username</i> is the username required to gain access to the server if necessary. <i>password</i> is the password for the specified username if required. <i>/directory</i> specifies the directory where the file is located if one exists. <i>/filename</i> specifies the name of the configuration file to be saved. NOTE: Configuration files should be named with a .cfg extension.
-noconfirm	<i>Optional:</i> Executes the command without displaying the "Are you sure? [Yes No]" confirmation prompt.
passive	Use passive mode FTP.

To copy a configuration file called *system.cfg* from a directory that was called *cfgfiles* to a directory named *configs_old* on the flash memory in the MIO/UMIO, enter the following command:

```
copy /flash/cfgfiles/system.cfg /flash/configs_old/system_2011.cfg
```

To copy a configuration file called *simple_ip.cfg* from a directory called *host_name_configs* to an FTP server with an IP address of *192.168.34.156*, on which you have an account with a username of *administrator* and a password of *secure*, use the following command:

```
copy /flash/host_name_configs/simple_ip.cfg
ftp://administrator:secure@192.168.34.156/host_name_configs/ simple_ip.cfg
```

To copy a configuration file called *init_config.cfg* to the root directory of a TFTP server with a hostname of *config_server*, enter the following command:

```
copy /flash/cfgfiles/init_config.cfg tftp://config_server/init_config.cfg
```

Deleting Files

The **delete** command removes a designated file from its specified location on the local file system.

 **Important:** This command does not support wildcard entries; each filename must be specified in its entirety.

 **Caution:** Do not delete the *boot.sys* file. If deleted, the system will not reboot on command and will be rendered inoperable.

```
delete { /flash | /usb1 | /hd-raid }/filename [-noconfirm]
```

Table 11. delete Command Variables

Keyword/Variable	Description
<i>filename</i>	The name of the file, including any extension, that will be deleted.
-noconfirm	Executes the command without displaying the “Are you sure? [Yes No]” confirmation prompt.

The following command deletes a file named *test.cfg* from the */flash* directory.

```
delete /flash/test.cfg
```

Removing Directories

The **rmdir** command deletes a current directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

 **Important:** The directory you want to remove (delete) must be empty before executing the **rmdir** command. If the directory is not empty, the CLI displays a “Directory not empty” message and will not execute.

```
rmdir url /dir_name
```

Table 12. rmdir Command Options

Keyword/Variable	Description
<i>url</i>	Specifies the path name of the local device. The <i>url</i> variable must be formatted as follows:
	{ /flash /usb1 /hd-raid } /<directory>/<file_name>
	file:/ { /flash /usb1 /hd-raid } /<directory>/<file_name>
<i>dir_name</i>	The name of the directory to be removed.
-noconfirm	Executes the command without displaying the “Are you sure? [Yes No]” confirmation prompt.

The following command deletes an empty directory named *configs* in the */flash* directory.

```
rmdir /flash/configs
```

Formatting Local Devices

The **format** command performs a low-level format of a local device. This operation formats the device to use the FAT16 formatting method, which is required for proper read/write functionality with the operating system.

 **Important:** Local devices that have been formatted using other methods such as NTFS or FAT32 may be used to store various operating system, CLI configuration, and crash log files. However, when placing a new local device into the MIO/UMIO for regular use, you should format the device via the system prior to use. This ensures that the proper file allocation table format is used, preventing any possible discrepancies between other formats used with other operating systems.

 **Caution:** The **format** command removes all files and information stored on the device.

To format a local device for use by the local file system, enter the following command:

```
format { /flash | /usb1 | /hd-raid }
```

Applying Pre-existing CLI Configuration Files

A pre-existing CLI configuration file is any .cfg file created to provide utility functions (such as clearing all statistics during testing) or created off-line using a text editor. There may be pre-existing configuration files stored on the local file system that can be applied to a running system at any time.

 **Caution:** If a configuration file is applied to a system currently running another CLI configuration, any like contexts, services, logical interfaces, physical ports, IP address pools, or other configured items will be overwritten if the same command exists in the configuration file being applied. Take caution to ensure that you are knowledgeable of the contents of the file being applied and understand what the service ramifications are if a currently running command is overwritten. Also note that changes will not be saved automatically.

A CLI configuration file, or script containing CLI commands, can be applied to a running system by entering the following command at the Exec mode prompt:

```
configure url [ verbose ]
```

Table 13. configure Command Options

Keyword/Variable	Description
<i>url</i>	Specifies the location of the CLI configuration file to be applied. The <i>url</i> may refer to a local or a remote file. The URL must be formatted according to one of the following formats:
	file:// { /flash /usb1 /hd-raid } /<directory>/<file_name>
	tftp:// { <ipaddress> <host_name> } /<directory>/<file_name>
	ftp:// [<username> [:<password>] @] <ipaddress> <host_name> /<directory>/<file_name>
	sftp:// [<username> [:<password>] @] <host> [/<path>] /<filename>
	http:// [<username> [:<password>] @] <host> [/<path>] /<filename>
	Where <i>username</i> is the user to be authenticated as, <i>password</i> is the password to use for authentication, <i>host</i> is the IP address using the standard IPv4 or the logical host name, and <i>path</i> is the directory structure to the file. <i>filename</i> is the name of the CLI configuration file or script that you wish to apply. The use of long filenames is supported.
verbose	Displays each line and its line number when applying a pre-existing CLI configuration file or script.

The following command applies a pre-existing CLI configuration file named *clearcmds.cfg* in the */flash* directory.

```
configure /flash/clearcmds.cfg
```

Viewing Files on the Local File System

This section describes how to view a variety of files.

Viewing the Contents of a Local Device

The contents, usage information, and file system directory structure of any local device can be viewed by entering the following command at the Exec mode prompt:

```
directory { /flash | /usb1 | /hd-raid }
```

Viewing CLI Configuration and boot.sys Files

The contents of CLI configuration and boot.sys files, contained on the local file system, can be viewed off-line (without loading them into the OS) by entering the following command at the Exec mode prompt:

```
show file url { /flash | /usb1 | /hd-raid } /filename
```

Where: *url* is the path name for the location of the file and *filename* is the name of the file, including any extension.



Important: Operator and inspector-level users can execute the **show file** command but cannot execute the **directory** command.

Validating an Operating System File

The operating system software image file, identified by its .bin extension, is a non-readable, non-editable file that executes on the system, creating its runtime operating system (OS).

It is important to verify a new operating system image file before attempting to load it. To accomplish this, a proprietary checksum algorithm is used to create checksum values for each portion of the application stored within the .bin file during program compilation.

This information can be used to validate the actual file against the checksum values stored within the file during its compilation. If any portion of the image file has become corrupted (for example, the file was truncated or was transferred using ASCII mode instead of binary mode), then this information is reported and the file is deemed unusable.

To validate an operating system software image file, enter the following command at the Exec mode prompt:

```
show version { /flash | /usb1 | /hd-raid } /[directory]/filename [all]
```

The output of this command displays the following information:

- Version – NN.N (build number)
- Description
- Date
- Boot Image
- Size
- Flags
- Platform – ASR5500

If an invalid file is found, the system displays a failure message similar to these:

```
Failure: Image /flash/os_3888.bin CRC check failed!
```

```
Failure: /flash/OS.3819.bin, has a bad magic number
```

Configuring the Boot Stack

The boot stack consists of a prioritized listing of operating system software image-to-CLI configuration file associations. These associations determine the software image and configuration file that gets loaded during system startup or upon a reload/reboot. Though multiple associations can be configured, the system uses the association with the highest priority. In the event that there is an error processing this association (for example, one of the files cannot be located), the system attempts to use the association with the next highest priority. Priorities range from 1 to 100, with 1 being the highest priority. The maximum number of boot stack entries that may be configured in the boot.sys file is 10.

Boot stack information is contained in the boot.sys file, described in [Understanding the boot.sys File](#). In addition to boot stack entries, the boot.sys file contains any configuration commands required to define the system boot method as explained in the section that follows.

System Boot Methods

The local-boot method uses software image and configuration files stored locally on the system. Upon system startup or reboot, the system looks on one of its local devices or **/hd-raid** located on the active MIO/UMIO for the specific software image and accompanying configuration text file. When using the local-booting method, you only need to configure boot stack parameters.

The system can also be configured to obtain its software image from a specific external network server while it is paired with a configuration text file that resides on the system. When using network booting, you need to configure the following:

- Boot stack parameters, which define the files to use and in what priority to use them
- Boot interface and network parameters defining the MIO/UMIO remote management LAN interface and the methods to use to reach the external network server
- Network booting delay time and optional name server parameters defining the delay period (in seconds) to allow for network communications to be established, and the IP address of any Domain Name Service (DNS) name server that may be used

Detailed information on how to configure the system to use the network booting method appears in [Network Booting Configuration Requirements](#).

Viewing the Current Boot Stack

To view the boot stack entries contained in the boot.sys file run the Exec mode **show boot** command.



Important: Operator and inspector-level users can execute the **show boot** command.

The example below shows the command output for a local booting configuration. Notice that in this example both the image file (operating system software) and configuration file (CLI commands) are located on the **/flash** device.

```
boot system priority 18 image /flash/build15003.aaaa.bin \config
/flash/general_config.cfg

boot system priority 19 image /flash/build14489.bbbb.bin \config
/flash/general_config_3819.cfg
```

```
boot system priority 20 image /flash/build14456.cccc.bin \config
/flash/general_config_3665.cfg
```

The example below shows the output for a combination network booting and local booting configuration. Notice in this example that the first two boot stack entries (Priorities 18 and 19) load the image file (operating system software) from an external network server using the Trivial File Transfer Protocol (TFTP), while all configuration files are located on the **/flash** device.

Also notice the boot network interface and boot network configuration commands located at the top of the boot stack. These commands define what MIO/UMIO remote management LAN interface(s) to use and information about communicating with the external network server that hosts the operating system software image file.

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0

boot delay 15

boot system priority 18 image tftp://192.168.1.161/tftpboot/build15003.st40.bin
\config /flash/general_config.cfg

boot system priority 19 image tftp://192.168.1.161/tftpboot/build14489.st40.bin
\config /flash/general_config.cfg

boot system priority 20 image /flash/build14456.st40.bin \config
/flash/general_config.cfg
```

To identify the boot image priority that was loaded at the initial boot time enter:

```
show boot initial-config
```

The example below displays the output:

```
[local]host# show boot initial-config
Initial (boot time) configuration:
  image tftp://192.168.1.161/tftpboot/build15429.xxxx.bin \
  config /flash/general_config.cfg
  priority 1
```

Adding a New Boot Stack Entry

 **Important:** Before performing this procedure, verify that there are less than 10 entries in the `boot.sys` file and that a higher priority entry is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Viewing the Current Boot Stack* for more information.

If priority 1 is in use, then you must renumber the existing entry(ies) to ensure that at least that priority is available. The maximum number of boot stack entries that can be contained in the `boot.sys` file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Refer to *Deleting a Boot Stack Entry* for more information.

This procedure details how to add new boot stack entries to the `boot.sys` file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
```

```
boot system priority number image <image_url> config <cfg_url>
```

Table 14. boot system priority Command Options

Keyword/Variable	Description
<i>number</i>	Specifies the boot priority number for the file group (combination of operating system software image and CLI configuration file). This value must be entered as an integer, ranging from 1 to 100, with the lowest number having the highest boot priority. An existing priority number, used by another boot stack entry, may be entered. However, this will overwrite the existing entry in the boot.sys file.
<i>image_url</i>	Specifies the location of the operating system software image file to be applied. The URL may refer to a local or a remote device. It must be formatted as follows:
	[file:] { /flash /usb1 /hd-raid } [/<path>] /<file_name>
	[http: ftp:] // <host> [:<port>] [/<path>] /<filename>
<i>cfg_url</i>	Specifies the name of the local device and directory path where the CLI configuration file is located. The URL must be formatted as follows: [file:] { /flash /usb1 /hd-raid } [/<path>] /<file_name> Where <i>path</i> is the directory structure to the file, and <i>file_name</i> is the name of the configuration file to load. This file typically has a .cfg extension.

The following command creates a new boot stack entry, using a boot priority of 3, an image file named *os_20000.XXX.bin*, and a configuration file named *general.cfg*.

```
boot system priority 3 image /flash/os_20000.XXX.bin config /flash/general.cfg
```



Important: Boot stack changes saved to the boot.sys file are not executed until the system is rebooted.

Synchronize the local file systems on the MIO/UMIOs with the following command:

```
fielssystem synchronize all
```

Deleting a Boot Stack Entry

This procedure details how to remove an individual boot stack entry from the boot.sys file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
```

```
no boot system priority number
```

Where *number* specifies the boot priority used for the boot stack entry. This command removes that specific entry from the boot stack, causing the boot.sys file to be overwritten.

Network Booting Configuration Requirements

Configuring the Boot Interface

Boot interface parameters define the MIO/UMIO management LAN interface that the system will use to communicate with the management network when using the network booting method.

This procedure details how to configure the boot interface for reliable communications with your network server. Make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Step 1 Enter the Global Configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the following command:

```
boot interface { local-eth1 | local-eth2 } medium { auto | speed { 10 | 100 |
1000 } duplex { full | half } } media { rj45 | sfp }
```

Keyword/Variable	Description
interface	Specifies the desired MIO/UMIO interface to use when communicating with the network server during boot. local-eth1 corresponds to the port 1 interface on the MIO/UMIO. local-eth2 interface that corresponds to the port 2 interface on the MIO/UMIO. NOTE: Use port 1 for network booting.
medium	Specifies the speed that the interface should implement to communicate on the network. auto implements auto-negotiation to determine the highest possible speed and duplex mode. speed specifies the rate to use as either 10 Mbps (10), 100 Mbps (100), or 1000 Mbps (1000). This command keyword must be following by the speed of the Ethernet connection, entered as an integer. NOTE: If the speed is manually configured, you must also configure the duplex mode. In addition, you must ensure that the network server configuration supports the speed and duplex configuration.
duplex	If the medium speed is manually configured, you must also configure the duplex mode through this parameter. Either full or half duplex mode can be implemented. NOTE: Ethernet networking rules dictate that if a device's interface is configured for auto-negotiation is communicating with a device that is manually configured to support full duplex, the first device will negotiate to the manually configured speed of the second device, but will only communicate in half duplex mode.
media	The media for MIO/UMIO port 1 is fixed at rj45 .

Step 3 Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring the Boot Network

Boot network parameters define the protocols and IP address information for MIO/UMIO interfaces used to reach the external network server that hosts the operating system software image file. To configure boot network parameters, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Step 1 Enter the Global Configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the following command:

```
boot networkconfig { dhcp | { { dhcp-static-fallback|static } ip address mio5
ip_address5 [ mio6 ip_address6 ] netmask subnet_mask [ gateway gw_ip_address ] }
}
```

Keyword/Variable	Description
dhcp	Specifies the use of the Dynamic Host Control Protocol (DHCP) to automatically assign an IP address to the interface at startup. NOTE: If this option is selected, you will not have to configure IP address information for the MIO/UMIO interface, defined using the boot interface command, or any needed gateway.
dhcp-static-fallback	Specifies the use of the DHCP to automatically assign an IP address to the MIO/UMIO interface, defined using the boot interface command, at startup. However, this option allows the configuration of a fallback static IP address that can be used when the DHCP server is unreachable.
static	Specifies that a static IP address will be configured for the MIO/UMIO interface, defined using the boot interface command.
ip address	If either the dhcp-static-fallback or static options were used as the method by which the MIO/UMIO interface obtains an IP address, then these keywords specify the static address.
mio5 <i>ip_address5</i>	Specifies the IP address to use for the MIO/UMIO interface in slot 5. Enter the <i>ip_address5</i> variable as an IP address in IPv4 dotted-decimal notation.
mio6 <i>ip_address6</i>	Specifies the IP address to use for the MIO/UMIO interface in slot 6. Enter the <i>ip_address6</i> variable as an IP address in IPv4 dotted-decimal notation. If used, both interfaces will appear in the boot.sys file.
netmask	Enter the subnet mask, using IPv4 dotted-decimal notation.
gateway	If either dhcp-static-fallback or static options were chosen as the method by which the interface will receive an IP address, this optional parameter specifies the IP address in IPv4 dotted-decimal notation for the next-hop gateway (router, bridge, etc.) to use, if needed.

The following command configures the boot network to communicate using DHCP, with a static-fallback IP address for MIO/UMIO in slot 5 of 192.168.206.101 and a Class C netmask.

```
boot networkconfig dhcp-static-fallback ip address mio5 192.168.206.101 netmask
255.255.255.0
```

The next example uses static IP addresses for MIO/UMIO in slot 5, which can access the external network server through a gateway whose IP address is 135.212.10.2.

```
boot networkconfig static ip address mio5 192.168.206.101 netmask 255.255.255.0
gateway 135.212.10.2
```

Step 3 Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring Boot Network Delay Time

An optional delay period, in seconds, can be configured for systems booting from a network. The purpose of this parameter is to allow time for external devices, such as switches, that use the Spanning Tree Protocol (STP) to determine the network route to a specified IP address.

To configure a boot network delay, enter the following command from the Global Configuration mode prompt.

```
boot delay time
```

Where *time* is an integer from 1 to 300 seconds before attempting to contact the external network server. If your network uses STP, a typical delay time of 30 seconds should suffice.



Important: Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring a Boot Nameserver

To enter the hostname of the network server that hosts the operating system software image, first configure the IP address of the Domain Name Service (DNS) server, referred to as a name server, that can resolve the host name for the machine.

To configure a boot nameserver address, enter the following command from the Global Configuration mode prompt.

```
boot nameserver ip_address
```

Where *ip_address* is the IP address of the DNS server entered in IPv4 dotted-decimal notation.



Important: Save the configuration as described in *Verifying and Saving Your Configuration*.

Upgrading the Operating System Software

The following information is required prior to performing a software upgrade:

- Current operating system version
- New operating system version
- Upgrade method

Identifying OS Release Version and Build Number

The operating system can be configured to provide services and perform pre-defined functions through commands issued from the CLI or through the Web Element Manager application.

The operating system software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system. Each software image can be identified by its release version and its corresponding build number. For example: `production.<build_number>.asr5500.bin` `production.<build_number>.asr5500.bin`.

The software version information can be viewed from the CLI in the Exec mode by entering the **show version** command.

```
[local<host_ame># show version
Active Software:
  Image Version:          15.0 (49328)
  Image Branch Version:   015.000 (001)
  Image Description:      Production_Build
  Image Date:             Tue Apr 23 00:45:12 EDT 2013
  Boot Image:            Unknown
```

Verify Free Space on the /flash Device

Verify that there is enough free space on the **/flash** device to accommodate the new operating system image file by entering the following Exec mode command:

```
[local]<host_name># directory /flash
```

The following is an example of the type of directory information displayed:

```
-rwxrwxr-x  1 root    root      7334 May  5 17:29 asr-config.cfg
-rwxrwxr-x  1 root    root       399 Jun  7 18:32 system.cfg
-rwxrwxr-x  1 root    root    10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x  1 root    root    10667 Jun  1 11:21 testconfig_4.cfg
-rwxrwxr-x  1 root    root     5926 Apr  7 16:27 tworpccontext.cfg
-rwxrwxr-x  1 root    root    15534 Aug  4 13:31 test_vlan.cfg
-rwxrwxr-x  1 root    root     2482 Nov 18 11:09 gateway2.cfg
-rwxrwxr-x  1 root    root 159106048 Dec 31 2011 production.41873.asr5000
1136352 /flash
Filesystem          1k-blocks      Used Available Use% Mounted on
/var/run/storage/flash/part1 31154688    1136352  30018336 4%
/mnt/user/.auto/onboard/flash
```

Note the “Available” blocks in the last line of the display. After displaying the directory information, the CLI returns to root and the following prompt appears:

```
[local]<host_name>#
```

Download the Software Image from the Support Site

Access to the Cisco support site and download facility is username and password controlled. You must have an active customer account to access the site and download the StarOS image.

Download the software image to a network location or physical device (USB stick) from which it can be uploaded to the **/flash** device.

Contact your Cisco representative or Cisco TAC for additional information.

Transfer StarOS Image to /flash on the Chassis

Transfer the new operating system image file to the **/flash** device on the MIO/UMIO using one of the following methods:

- Copy the file from a network location or local device plugged in into the MIO/UMIO by entering the following command:

```
[local]<host_name># copy <from_url> <to_url> [-noconfirm]
```

- Transfer the file to the **/flash** device using an FTP client with access to the system.



Important: Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable.

- Transfer the file to the **/flash** device using an SFTP client with access to the system.

Verify that the image file was successfully transferred to the **/flash** device by running the following Exec mode command:

```
[local]<host_name># directory /flash
```

The `production.<build_number>.asr5500.bin` image filename should appear in the displayed output.

Run the `show version /flash/<image_filename>` command to verify the build information. For example:

```
[local]ASR55# show version /flash/production.43582.asr5000.bin
[local]ASR55# show version /flash/production.43582.asr5500.bin
Active Software:
OPERATIONAL_IMAGE Version      : 14.0 (47890)
OPERATIONAL_IMAGE Description  : ASR5500 Production Build <47890>
OPERATIONAL_IMAGE Date        : Friday May 25 20:17:24 GMT 2012
OPERATIONAL_IMAGE Size        : 169981952
OPERATIONAL_IMAGE Flags       : None
OPERATIONAL_IMAGE Platform    : ASR5500
```

Saving a Copy of the Current Configuration File

Prior to upgrading to a new software release, you should copy and rename the current configuration file to the **/flash** device and to an off-chassis location (external memory device or network URL). This renamed copy assures that you will have a fallback, loadable configuration file should a problem be encountered during the upgrade.

Off-line Software Upgrade

An off-line software upgrade can be performed for any system, upgrading from any version of operating system software to any version, regardless of version number. This process is considered off-line because while many of the steps can be performed while the system is currently supporting sessions, the last step of this process requires a reboot to actually apply the software upgrade.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Configure a Newcall Policy

Configure a newcall policy from the Exec mode to meet your service requirements. When enabled the policy redirects or rejects new calls in anticipation of the chassis reload that completes the upgrade process. This reduces the amount of service disruption to subscribers caused by the system reload that completes the upgrade.

 **Important:** Newcall policies are created on a per-service basis. If you have multiple services running on the chassis, you can configure multiple newcall policies.

The syntax for newcall policies is described below:

```
newcall policy { asngw-service | asnpc-service | sgsn-service } { all | name
service_name } reject

newcall policy cscf-service { all | name service_name } { redirect
target_ip_address [ weight weight_num ] [ target_ipaddress2 [ weight weight_num ]
... target_ip_address16 [ weight weight_num ] | reject }

newcall policy { fa-service | lns-service | mipv6ha-service } { all | name
service_name } reject

newcall policy { ha-service | pdsn-service } { all | name service_name } {
redirect target_ip_address [ weight weight_num ] [ target_ipaddress2 [ weight
weight_num ] ... target_ip_address16 [ weight weight_num ] |reject}

newcall policy ggsn-service {apn name apn_name | all | name service_name}reject

newcall policy hnbgw-service {all | name service_name}reject

newcall policy {pcc-af-service | pcc-policy-service} {all | name
service_name}reject
```

```
newcall policy {pcc-af-service | pcc-policy-service} {all | name
service_name}reject
```

```
newcall policy mme-service {all | name service_name } reject
```

Keyword/Variable	Description
name <service_name> name <apn_name>	Specifies a single instance of a service type or an APN to apply the newcall policy to. <i>service_name</i> is the name of a service that was previously configured. It can consist of up to 63 alphanumeric characters and is case sensitive. <i>apn_name</i> is the name of a previously configured APN expressed as an alphanumeric string of 1 through 63 characters that is case sensitive. To apply the newcall policy to a subset of all of the configured services of a specific type, re-issue the command for each individual service name desired.
redirect <target_ipaddress>	Configures the busy-out action. When a redirect policy is invoked, the service rejects new sessions and provides the IP address of an alternate destination. This command can be issued multiple times. The IP address of an alternate destination expressed in IPv4 dotted-decimal format. Up to 16 IP addresses can be specified either in one command or by issuing the redirect command multiple times. If you try to add more than 16 IP addresses to the redirect policy the CLI issues an error message. If you specify an IP address and weight that already exists in the redirect policy the new values override the existing values Depending on the type of service that the policy is applied to, a reason code is reported as part of the reply to indicate the rejection reason.
weight <i>weight_num</i>	When multiple addresses are specified, they are selected in a weighted round-robin scheme. Addresses with higher weights are more likely to be selected when redirecting traffic. If a weight is not specified the entry is automatically assigned a weight of 1. <i>weight_num</i> must be an integer from 1 through 10.
reject	Specifies that the policy will reject new incoming session requests.

Configure a “Message of the Day” Banner

Optional: Configure a “Message of the Day” banner informing other management users that the system will be rebooted by entering the following command from the Global Configuration mode prompt.

```
banner motd " banner_text"
```

banner_text is the message that you would like to be displayed and can be up to 2048 alphanumeric characters. Note that *banner_text* must begin with and end in quotation marks (“”). For more information in entering CLI banner information, see the *CLI Reference*. The banner is displayed when an administrative user logs onto the CLI.

Back up the Current CLI Configuration File

Back up the current CLI configuration file by entering the following command:

```
[local]<host_name># copy from_url to_url [-noconfirm]
```

This creates a mirror-image of the CLI configuration file linked to the operating system defined in the current boot stack entry.

The following command example creates a backup copy of a file called `general.cfg` located on the `/flash` device to a file called `general_3652.cfg`:

```
[local]<host_name># copy /flash/general.cfg /flash/general_3652.cfg
```

Create a New Boot Stack Entry

Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:

```
[local]<host_name>(config)# boot system priority <number> image <image_url>
/flash/<filename> config <cfg_url> /flash/<filename>
```

Assign the next highest priority to this entry, by using the `<N-1>` method, wherein you assign a priority number that is one number less than your current highest priority.

 **Important:** Run the Exec mode `show boot` command to verify that there are less than 10 entries in the `boot.sys` file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the `boot.sys` file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the `no boot system priority` command to delete a boot stack entry.

```
[local]<host_name># configure
[local]<host_name>(config)# no boot system priority <number>
```

To add new boot stack entries to the `boot.sys` file enter the following commands:

```
[local]<host_name># configure
[local]<host_name>(config)# boot system priority <number> image <image_url>
config <cfg_url>
```

For information on using the `boot system priority` command, refer to the *Adding a New Boot Stack Entry* section.

Synchronize File Systems

Synchronize the local file systems on the management cards by entering the following command:

```
filesystem synchronize all
```

Reboot the Chassis

Reboot the chassis by entering the following command:

```
[local]<host_name># reload [-noconfirm]
```

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

After the system reboots, establish a CLI session and enter the **show version** command to verify that the active software version is correct.

Optional for PDSN: If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

Verify the Running Software Version

After the system has successfully booted, verify that the new StarOS version is running by executing the Exec mode **show version** command. For example:

```
[local<host_ame># show version
Active Software:
  Image Version:          15.0 (49328)
  Image Branch Version:   015.000(001)
  Image Description:     Production_Build
  Image Date:            Tue Apr 23 00:45:12 EDT 2013
  Boot Image:            Unknown
```

Restoring the Previous Software Image

If for some reason you need to undo the upgrade, perform the upgrade again except:

- Specify the locations of the upgrade software image and configuration files.
then
- Specify the locations of the original software image and configuration files.

Upgrading ICSR Chassis

The procedure for upgrading primary and backup ICSR chassis is described in *Interchassis Session Recovery*. Essentially the procedure requires upgrading the primary and standby chassis using the off-line method while each is in standby mode.

Performing Dynamic Software Updates

This section describes the dynamic software update (DSU) process that can be used to incrementally update plugins without having to update StarOS and reload the system.

Overview

StarOS allows the runtime loading of plugins. All StarOS builds include a “default” baseline plugin.

This feature is currently used to dynamically update the detection logic used to filter P2P applications.

Patching is the process used to install a plugin as an incremental update to a StarOS release. One plugin can be provided to multiple, compatible, concurrent product releases. A plugin is distributed in the form of a compressed distribution kit via the internet or by other means (USB stick, CD, etc.).

A plugin is a functional software entity that provides incremental updates to a pre-existing StarOS software component. Plugins have the characteristic of being dynamically loadable at runtime and do not require a system restart. A plugin has a name and one or more versions. All plugin names are known to the system at product release.

A plugin module is a specific instance of a plugin version consisting of at least one file that can be added to a running, in-service system. The module contains the information or instructions for a specific component's incremental update. Typically this would be a single file.

The Version Priority List (VPL) is a linked list of module versions associated with a specific plugin. Each plugin has one VPL. The list is sorted in ascending order by the priority number that is assigned by the administrator. When updating, the lowest priority number is tried first. If that version is not successful, the version in the VPL with the next sequentially greater priority number is tried. This list is iterated until a successful version is found. The VPL also supports manual rollback to a previous version (higher priority number) via a CLI command.

The basic sequence for the dynamic software process is as follows:

- [Downloading the Patch Kit](#)
- [Unpacking the Patch Kit](#)
- [Configuring the Plugin](#)
- [Loading the Plugin](#)
- [Rolling Back to a Previous Plugin Version](#)

Downloading the Patch Kit

The Exec mode **patch plugin** command copies a patch intended for a specific plugin onto the running system.

The plugin kit includes a compressed plugin file (.tgz extension) and a certificate file (.cert extension). The command syntax is as follows:

```
patch plugin <name> [ http | ftp | sftp ]://<host>/<directory>/<file> certificate  
[ http | ftp | sftp ]://<host>/<directory>/<file>
```

For example:

```
[local]asr5k# patch plugin p2p http://192.168.1.2/tmp/patch_libp2p-1.17.4343.tgz
certificate http://192.168.1.2/tmp/patch_libp2p-1.17.4343.cert
New patch for plugin p2p available for installation
```

The patch kit is copied to the **/flash/patch/<module-name>** directory on the system.

Unpacking the Patch Kit

The Exec mode **install patch** command unpacks a patch kit and validates its contents. This command does not actually distribute the contents to the packet processing card **/var/opt** directories

The command syntax is as follows:

```
install plugin <plugin-name> <patch-file-name>
```

For example:

```
[local]asr5# install patch plugin p2p patch_libp2p-1.17.4343.tgz
Install patch file patch_libp2p-1.17.4343.tgz [Yes/No]?
Install module p2p version 1.17.4343 successful
```

When a new plugin module is installed, the patch file is checked against the certificate which is provided with the patch file for authenticity of contents. If the certificate matches the patch file, the contents are extracted to the **/flash/install/<plugin-name>/contents** directory.

Once the contents are extracted, the Plugin Manager looks for the file **patch.bom** which contains a list of files which are considered a complete set for the patch. In addition, for each file listed, the bom file contains an intended final destination directory where the content file should reside at the end of the installation.

For each file listed in the bom file, the installation procedure copies the specified file to the destination directory. Once this has completed, the installation procedure is finished. The primary MIO/UMIO is responsible for ensuring that the **/flash** partitions on the standby MIO/UMIO are up-to-date.

Configuring the Plugin

The Global Configuration mode **plugin** command specifies a plugin to be configured and enters the Plugin Configuration mode. The plugin name must match the name of a plugin which has been copied to and unpacked on the system or an error message is displayed.

The command syntax is as follows:

```
plugin <plugin-name>

module priority <number> version <module_version>
```

For example:

```
[local]asr5# configure
[local]asr5(config)# plugin p2p
[local]asr5(plugin-p2p)# module priority 1 version 1.17.4343
```

If a module priority number (1 through 100) was specified in an earlier configuration, that number is preserved. If the user does not explicitly replace or remove a module priority number, that number and the version associated with it will remain configured.

Loading the Plugin

The Exec mode **update module** command loads the specified module with the lowest priority number from the VPL.

The command syntax is as follows:

```
update module <plugin-name>
```

For example:

```
[local]asr5k# update module p2p  
Update to module p2p version 1.17.4343 successful
```

The **update module** command takes the configured VPL (/flash/module.sys) and copies it into the internal data structures within the Plugin Manager. The VPL determines the sequence of module versions to try to load for each of the cards.

Current subscriber sessions will not be updated to the latest plugin version. New subscriber sessions will be affected by the newly updated plugin version.

Rolling Back to a Previous Plugin Version

The Exec mode **rollback module** command loads a module with the next higher priority number from the VPL.

The command syntax is as follows:

```
rollback module <plugin-name>
```

For example:

```
[local]asr5# rollback module p2p  
Rollback to module p2p version 1.17.4340 successful
```

Current subscriber sessions will continue to be affected by the previous plugin version. New subscriber sessions will be affected by the newly downgraded plugin version.

DSU show Commands

There are two **show** commands associated with the dynamic software update process – **show module** and **show plugin**.

The **show module** command displays the current status of the internal VPL. The command syntax is as follows:

```
show module [<plugin-name>]
```

For example:

```
[local]asr5# show module p2p
Module p2p
Priority version  loaded location                time status
1      1.17.4343    no  /var/opt/lib    07/04/2012:05:06:59  success
2      1.17.4340    yes /var/opt/lib    03/06/2012:04:05:53  success
3      1.17.4335    no  /var/opt/lib    03/05/2012:03:04:35  success
4      1.17.4328    no  /var/opt/lib    03/04/2012:02:03:42  failed
X      1.17.4289    no  /var/opt/lib    01/12/2012:05:44:22  success
```



Important: Priority “X” identifies the “default” module that shipped with the StarOS build.

The **show plugin** command displays the current configuration of the plugin – AVPs and version priorities. The command syntax is as follows:

```
show plugin [<plugin-name>]
```

For example:

```
[local]asr5# show plugin p2p
plugin p2p
module priority 1 version 1.17.4343
module priority 2 version 1.17.4340
module priority 3 version 1.17.4328
module priority 4 version 1.17.4340
module priority X version 1.17.4289
```

Managing License Keys

License keys define capacity limits (number of allowed subscriber sessions) and available features on your system. Adding new license keys allows you to increase capacity and add new features as your subscriber base grows.

New System License Keys

New systems are delivered with no license keys installed. In most cases, you receive the license key in electronic format (usually through e-mail).

When a system boots with no license key installed a default set of restricted session use and feature licenses is installed. The following Exec Mode command lists the license information:

```
show license information
```

 **Important:** With no license key installed, the session use licenses for PDSN, HA, GGSN, and L2TP LNS are limited to 10,000 sessions.

The license keys on the ASR 5500 are stored in EEPROM on the chassis midplane. Both MIO/UMIOs access this EEPROM when booting.

Session Use and Feature Use Licenses

Session use and feature use licenses are software mechanisms that provide session limit controls and enable special features within the system. These electronic licenses are stored in the system's configuration file that is loaded as part of the system software each time the system is powered on or restarted.

- Session use licenses limit the number of concurrent sessions that a system is capable of supporting per service type and are acquired on an as-needed basis. This allows carriers to pay only for what they are using and easily increase capacity as their subscriber base grows.
- Feature use licenses enable specific features/functionality within the system and are distributed based on the total number of sessions supported by the system.

Installing New License Keys

Use the instructions below to install a new license key.

Cutting and Pasting the Key

If you have a copy of the license, use the following configuration to cut and paste just the license key part:

Step 1 From the Exec mode, enter the following:

```
configure
license key license
exit
```

license is the license key string. The license can be an alphanumeric string of 1 through 1023 characters that is case sensitive. Copy the license key as shown in the example below, including the “\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-
1|C2S=\STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|000000000
0000|LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|F
MG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd638
8jHsHD99sg36SG267gshssja77
end
```

Step 2 Verify that the license key just entered was accepted by entering the following command at the Exec mode prompt:

```
show license key
```

The new license key should be displayed. If it is not, return to the Global configuration mode and re-enter the key using the **license key** command.

Step 3 Verify that the license key enabled the correct functionality by entering the following command:

```
show license information
```

All license keys and the new session capacity or functionality enabled should be listed. If the functionality or session capacity enabled by the new key is incorrect, please contact your service representative.

Step 4 Save your configuration as described in *Verifying and Saving Your Configuration*.



Caution: Failure to save the new license key configuration in the current CLI configuration file will result in the loss of any of the new features enabled by the license key once the system is reloaded.

Adding License Keys to Configuration Files

License keys can be added to a new or existing configuration file.



Important: License key information is maintained as part of the CLI configuration. Each time a key is installed or updated, you must re-save the configuration file.

Step 1 Open the configuration file to which the new license key commands are to be copied.

Step 2 Copy the license as shown in the example, including the “\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-
1|C2S=\STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|000000000
0000|LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|F
MG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd638
8jHsHD99sg36SG267gshssja77
end
```

Step 3 Paste the license key into the configuration

 **Important:** Paste the license key information at the beginning of the configuration file to ensure the system has the expected capacity and features before it configures contexts.

Step 4 Save your configuration as described in *Verifying and Saving Your Configuration*.

License Expiration Behavior

When a license expires, there is a built-in grace period of 30 days that allows normal use of the licensed session use and feature use licenses. This allows you to obtain a new license without any interruption of service.

The following Exec mode command lists the license information including the date the grace period is set to expire:

```
show license information
```

Requesting License Keys

License keys for the system can be obtained through your Cisco account representative. Specific information is required before a license key may be generated:

- Sales Order or Purchase Order information
- Desired session capacity
- Desired functionality
- Midplane (chassis) serial number

To obtain the ASR 5500 chassis serial number, at the Exec mode prompt enter the `show card hardware 5` command. Look under the “MEC” heading for the “UDI Serial Number” as shown in the example below:

```
MEC:
  Description           : MEC
  Cisco Part Number     : 73-14501-01 A0
  UDI Serial Number     : FLM154300D8
  UDI Product ID       : ASR55-MEC
  UDI Version ID       : V01
```

Viewing License Information

To see the license detail, enter the following command from the Exec mode:

```
show license information [ full | key [ full ] ]
```

Deleting a License Key

Use the procedure below to delete the session and feature use license key from a configuration. You must be a security administrator or administrator.

```
configure
  no license key
exit
show license key
```

The output of this command should display: “No license key installed”.

Management Card Replacement and License Keys

License keys are stored on a midplane EEPROM in the ASR 5500 chassis. The MIO/UMIOs share these license keys. There is no need to swap memory cards into replacement MIO/UMIOs.

Managing Local-User Administrative Accounts

Unlike context-level administrative accounts which are configured via a configuration file, information for local-user administrative accounts is maintained in a separate file in flash memory and managed through the software's Shared Configuration Task (SCT). Because local-user accounts were designed to be compliant with ANSI T1.276-2003, the system provides a number of mechanisms for managing these types of administrative user accounts.

Configuring Local-User Password Properties

Local-user account password properties are configured globally and apply to all local-user accounts. The system supports the configuration of the following password properties:

- **Complexity:** Password complexity can be forced to be compliant with ANSI T1.276-2003.
- **History length:** How many previous password versions should be tracked by the system.
- **Maximum age:** How long a user can use the same password.
- **Minimum number of characters to change:** How many characters must be changed in the password during a reset.
- **Minimum change interval:** How often a user can change their password.
- **Minimum length:** The minimum number of characters a valid password must contain.

Refer to the `local-user password` command in *Global Configuration Mode Commands* in the *Command Line Interface Reference* for details on each of the above parameters.

Configuring Local-User Account Management Properties

Local-user account management includes configuring account lockouts and user suspensions.

Local-User Account Lockouts

Local-user accounts can be administratively locked for the following reasons:

- **Login failures:** The configured maximum login failure threshold has been reached. Refer to the `local-user max-failed-logins` command in *Global Configuration Mode Commands* in the *Command Line Interface Reference* for details.
- **Password Aging:** The configured maximum password age has been reached. Refer to the `local-user password` command in *Global Configuration Mode Commands* in the *Command Line Interface Reference* for details.

Accounts that are locked out are inaccessible to the user until either the configured lockout time is reached (refer to the `local-user lockout-time` command in *Global Configuration Mode Commands* in the *Command Line Interface Reference*) or a security administrator clears the lockout (refer to the `clear local-user` command in *Exec Mode Commands* in the *Command Line Interface Reference*).

 **Important:** Local-user administrative user accounts could be configured to enforce or reject lockouts. Refer to the `local-user username` command in *Global Configuration Mode Commands* in the *Command Line Interface Reference* for details.

Local-User Account Suspensions

Local-user accounts can be suspended as follows:

```
configure
suspend local-user <name>
```

A suspension can be removed by entering:

```
configure
no suspend local-user <name>
```

Changing Local-User Passwords

Local-user administrative users can change their passwords using the **password change** command in the Exec mode. Users are prompted to enter their current and new passwords.

Security administrators can reset passwords for local-users by entering the following command from the root prompt in the Exec mode:

```
password change username name
```

name is the name of the local-user account for which the password is to be changed. When a security administrator resets a local-user's password, the system prompts the user to change their password the next time they login.

All new passwords must adhere to the password properties configured for the system.

Chapter 9

Monitoring the System

This segment provides information for monitoring system status and performance using the **show** commands found in the Command Line Interface (CLI). These commands have many related keywords that allow them to provide useful information on all aspects of the system ranging from current software configuration through call activity and status.

The selection of keywords described in this segment is intended to provide the most useful and in-depth information for monitoring the system. For additional information on these and other **show** command keywords, refer to *Exec Mode show Commands* in the *Command Line Interface Reference*.

This segment includes the following sections:

- [SNMP Notifications](#)
- [Monitoring System Status and Performance](#)
- [Monitoring ASR 5500 Hardware Status](#)

SNMP Notifications

In addition to the CLI, the system supports Simple Network Management Protocol (SNMP) notifications that indicate status and alarm conditions. Refer to the *SNMP MIB Reference* for a detailed listing of these notifications.

Monitoring System Status and Performance

This section contains commands used to monitor the status of tasks, managers, applications and other software components in the system. Output descriptions for most of the commands are located in the *Statistics and Counters Reference*.

Table 15. System Status and Performance Monitoring Commands

To do this:	Enter this command:
View Administrative Information	
Display Current Administrative User Access	
View a list of all administrative users currently logged on the system	<code>show administrators</code>
View the context in which the administrative user is working, the IP address from which the administrative user is accessing the CLI, and a system generated ID number	<code>show administrators session id</code>
View information pertaining to local-user administrative accounts configured for the system	<code>show local-user verbose</code>
View statistics for local-user administrative accounts	<code>show local-user statistics verbose</code>
View information pertaining to your CLI session	<code>show cli</code>
Determining System Uptime	
View system uptime (time since last reboot)	<code>show system uptime</code>
View NTP Server Status	
View NTP servers status	<code>show ntp status</code>
View System Resources	
View all system resources such as CPU resources and number of managers created	<code>show resources [cpu]</code>
View System Alarms	
View information about all currently outstanding alarms	<code>show alarm outstanding all verbose</code>
View system alarm statistics	<code>show alarm statistics</code>
View Congestion-Control Statistics	
View Congestion-Control Statistics	<code>show congestion-control statistics</code>

To do this:	Enter this command:
View Remote Management Statistics	
Display SNMP Notification Statistics	
View SNMP notification statistics	<code>show snmp notifies</code>
Display SNMP Access Statistics	
View SNMP access statistics	<code>show snmp accesses</code>
Display SNMP Trap History	
View SNMP trap history	<code>show snmp trap history</code>
Display SNMP Trap Statistics	
View SNMP Trap Statistics	<code>show snmp trap statistics</code>
Display ORBEM Information	
View ORBEM client status	<code>show orbem client id</code>
View ORBEM session information	<code>show orbem session table</code>
View individual ORBEM sessions	<code>show orbem session id orbem</code>
View ORBEM status information	<code>show orbem status</code>
View Port Counters	
Display Port Datalink Counters	
View datalink counters for a specific port	<code>show port datalink counters slot#/port#</code>
Display Port Network Processor Unit (NPU) Counters	
View NPU counters for a specific port	<code>show port npu counters slot#/port#</code>



Important: The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s). Some commands have different outputs depending on the platform type.

Clearing Statistics and Counters

It may be necessary to periodically clear statistics and counters in order to gather new information. The system provides the ability to clear statistics and counters based on their grouping (PPP, MIPHA, MIPFA, etc.).

Statistics and counters can be cleared using the CLI **clear** command. Refer to *Exec Mode Commands* in the *Command Line Interface Reference* for detailed information on using this command.

Monitoring ASR 5500 Hardware Status

Use the commands contained in this section to monitor the status of the hardware components in the chassis. For output descriptions for most of the commands, refer to the *Statistics and Counters Reference*.



Important: The commands or keywords and variables are dependent on platform type, product version, and installed license(s). Some commands produce different outputs, depending on the platform type.

Table 16. Hardware Monitoring Commands

To do this:	Enter this command:
View the Status of the Power System	
View the status of the PFUs	<code>show power chassis</code>
View the power status of the individual chassis slots	<code>show power all</code>
View the Status of the Fan Trays	
View the status of the fan trays, including current relative speeds and temperatures.	<code>show fans</code>
Determine the Status of Installed Cards	
View a listing of installed application cards	<code>show card table</code>
Perform a Hardware Inventory	
View all cards installed in the chassis and their hardware revision, part, serial, assembly, and fabrication numbers	<code>show hardware inventory</code>
View details of a specific card. Output contains same information as output of both show hardware inventory and show hardware version board	<code>show hardware card slot_number</code>
View Card Diagnostics	
View boot, power and temperature diagnostics	<code>show card diag slot_number</code>
View runtime, or real time, information	<code>show card info slot_number</code>
View the LED Status of All Installed Cards	
View the LED status for all installed cards	<code>show leds all</code>
View Available Physical Ports	
View ports that are available to the system	<code>show port table</code>
View detailed information for a specific port	<code>show port info slot_number/port_number</code>

To do this:	Enter this command:
View CPU Resource Information	
View CPU resource information	<code>show resource cpu</code>
View CPU resources	<code>show resources { cpu session }</code>
View CPU usage information	<code>show cpu table; show cpu info</code>
View Component Temperature Information	
View current component temperatures	<code>show temperature</code>
View maximum temperatures reached since last timestamp.	<code>show maximum-temperatures</code>

Chapter 10

Bulk Statistics

This segment provides configuration information for:

- [Configuring Communication With the Collection Server](#)
- [Viewing Collected Bulk Statistics Data](#)
- [Manually Gathering and Transferring Bulk Statistics](#)
- [Clearing Bulk Statistics Counters and Information](#)
- [Bulk Statistics Event Log Messages](#)

Configuring Communication with the Collection Server

Two configuration methods are available for defining how bulk statistics are collected and managed. A “standard” configuration allows the system to automatically assign a number to the bulk statistics file. Optionally, a number can be specified by an administrator in the optional configuration method. Command details and descriptions of keywords and variables for commands in this segment are located in *Bulk Statistics Configuration Mode Commands* and *Bulk Statistics File Configuration Mode Commands* in the *Command Line Interface Reference*.

Configuring Standard Settings

The configuration example in this section defines basic operation of the bulk statistics feature. Use the following example configuration to set up the system to communicate with the statistic collection server:

```
configure
  bulkstats mode
    schema <name> format <format_string>
    sample-interval <time_interval>
    transfer-interval <xmit_time_interval>
    limit <mem_limit>
  exit
bulkstats collection
end
```

Configuring Optional Settings

This section describes optional commands that can be used within the Bulk Statistics Configuration mode. Specifically, bulk statistic “files” under which to group the bulk statistic configuration are configured using commands in this section. “Files” are used to group bulk statistic schema, delivery options, and receiver configuration. Because multiple “files” can be configured, this functionality provides greater flexibility in that it allows you to configure different schemas to go to different receivers.

```
configure
  bulkstats mode
    file <number>
      receiver <ip_address> { primary | secondary } [ mechanism { { { ftp |
sftp } login <user_name> [ encrypted ] password <pwd> } | tftp } } ] }
      receiver mode { redundant | secondary-on-failure }
```

```

    remotefile format <naming_convention> [ both-receivers | primary-
receiver | secondary-receiver ]

    header format <header_format>

    footer format <footer_format>

    exit

<name> schema format <format_string>

sample-interval <time_interval>

transfer-interval <xmit_time_interval>

limit <mem_limit>

exit

bulkstats collection

end

```

Configuring Bulk Statistic Schemas

In each configuration example described in *Configuring Standard Settings* and *Configuring Optional Settings*, the following is the primary command used to configure the type of schema and the statistics collected:

```
name schema format format_string
```

Refer to *Bulk Statistics Configuration Mode Commands* and *Bulk Statistics File Configuration Mode Commands* in the *Command Line Interface Reference* for more information regarding supported schemas, available statistics, and proper command syntax.

Verifying Your Configuration

After configuring support for bulk statistics on the system, you can check your settings prior to saving them.

Follow the instructions in this section to verify your bulk statistic settings. These instructions assume that you are at the root prompt for the Exec mode.

Check your collection server communication and schema settings by entering the following command:

```
show bulkstats schemas
```

The following is an example command output:

```

Bulk Statistics Server Configuration:
  Server State:           Enabled
  File Limit:             6000 KB
  Sample Interval:       15 minutes (0D 0H 15M)
  Transfer Interval:     480 minutes (0D 0H 15M)
  Collection Mode:       Cumulative
  Receiver Mode:        Secondary-on-failure

```

```

Local File Storage:          None
Bulk Statistics Server Statistics:
Records awaiting transmission: 114
Bytes awaiting transmission:  8092
Total records collected:      59926
Total bytes collected:        4190178
Total records transmitted:    59812
Total bytes transmitted:      4188512
Total records discarded:      0
Total bytes discarded:        0
Last collection time required: 2 second(s)
Last transfer time required:  0 second(s)
Last successful transfer:     Wednesday December 7 12:14:30 EDT 2011
Last successful tx recs:      190
Last successful tx bytes:     13507
Last attempted transfer:      Wednesday December 7 12:14:30 EDT 2011

File 1
Remote File
Format:          /users/ems/server/data/chicago/bulkstat%date%%time%.txt
File Header:     "CHI_test %time%"
File Footer:     ""
Bulkstats Receivers:
Primary: 192.168.0.100 using FTP with username administrator
Records awaiting transmission: 0
Bytes awaiting transmission:  0
Total records collected:      0
Total bytes collected:        0
Total records transmitted:    0
Total bytes transmitted:      0
Total records discarded:      0
Total bytes discarded:        0
Last transfer time required:  0 second(s)
No successful data transfers
No attempted data transfe

File 2 not configured

File 3 not configured

File 4 not configured

```

Saving Your Configuration

Save the configuration as described in *Verifying and Saving Your Configuration*.

Viewing Collected Bulk Statistics Data

The system provides a mechanism for viewing data that has been collected but has not been transferred. This data is referred to as “pending data”.

View pending bulk statistics data per schema by entering the following:

```
show bulkstats data
```

The above command also shows the statistics of remote files, if configured as described in [Configuring Optional Settings](#).

The following is a sample output:

```
Bulk Statistics Server Statistics:
Records awaiting transmission: 1800
Bytes awaiting transmission: 163687
Total records collected: 1800
Total bytes collected: 163687
Total records transmitted: 0
Total bytes transmitted: 0
Total records discarded: 0
Total bytes discarded: 0
Last collection time required: 2 second(s)
Last transfer time required: 0 second(s)
No successful data transfers
Last attempted transfer: Monday February 14 15:12:30 EST 2011

File 1
Remote File Format: %date%%time%
File Header: "Format 4.5.3.0"
File Footer: ""
Bulkstats Receivers:
Primary: 192.168.1.200 using FTP with username root
File Statistics:
Records awaiting transmission: 1800
Bytes awaiting transmission: 163687
Total records collected: 1800
Total bytes collected: 163687
Total records transmitted: 0
Total bytes transmitted: 0
Total records discarded: 0
Total bytes discarded: 0
Last transfer time required: 0 second(s)
No successful data transfers
Last attempted transfer: Monday February 14 15:12:30 EST 2011

File 2 not configured

File 3 not configured

File 4 not configured
```

Manually Gathering and Transferring Bulk Statistics

There may be times where it is necessary to gather and transfer bulk statistics outside of the scheduled intervals. The system provides commands that allow you to manually initiate the gathering and transferring of bulk statistics.

These commands are issued from the Exec mode.

To manually initiate the gathering of bulk statistics outside of the configured sampling interval, enter the following command:

```
bulkstats force gather
```

To manually initiate the transferring of bulk statistics prior to reaching the of the maximum configured storage limit, enter the following command:

```
bulkstats force transfer
```

Clearing Bulk Statistics Counters and Information

It may be necessary to periodically clear counters pertaining to bulk statistics in order to gather new information or to remove bulk statistics information that has already been collected. The following command can be used to perform either of these functions:

```
clear bulkstats { counters | data }
```

Keyword/Variable	Description
counters	Clears the counters maintained by the system “bulkstats” facility.
data	Clears any accumulated data that has not been transferred. This includes any "completed" files that have not been successfully transferred.

Bulk Statistics Event Log Messages

The stat logging facility captures several events that can be useful for diagnosing errors that could occur with either the creation or writing of a bulk statistic data set to a particular location.

The following table displays information pertaining to these events.

Table 17. Logging Events Pertaining to Bulk Statistics

Event	Event ID	Severity	Additional Information
Local File Open Error	31002	Warning	"Unable to open local file <i>filename</i> for storing bulkstats data"
Receiver Open Error	31018	Warning	"Unable to open url <i>filename</i> for storing bulkstats data"
Receiver Write Error	31019	Warning	"Unable to write to url <i>filename</i> while storing bulkstats data"
Receiver Close Error	31020	Warning	"Unable to close url <i>filename</i> while storing bulkstats data"

Chapter 11

System Logs

This segment describes how to configure parameters related to the various types of logging and how to viewing their content. It includes the following sections:

- [Configuring Event Logging Parameters](#)
- [Configuring Active Logs](#)
- [Specifying Facilities](#)
- [Configuring Trace Logging](#)
- [Configuring Monitor Logs](#)
- [Viewing Logging Configuration and Statistics](#)
- [Viewing Event Logs Using the CLI](#)
- [Configuring and Viewing Crash Logs](#)
- [Saving Log Files](#)
- [Event ID Overview](#)

System Log Types

There are five types of logs that can be configured and viewed on the system:



Important: Not all Event Logs can be configured on all products. Configurability depends on the hardware platform and licenses in use.

- **Event:** Event logging can be used to determine system status and capture important information pertaining to protocols and tasks in use by the system. This is a global function that will be applied to all contexts, sessions, and processes.
- **Active:** Active logs are operator configurable on a CLI instance-by-CLI instance basis. Active logs configured by an administrative user in one CLI instance cannot be viewed by an administrative user in a different CLI instance. Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as events are generated.
- **Trace:** Trace logging can be used to quickly isolate issues that may arise for a particular connected subscriber session. Traces can be taken for a specific call identification (callid) number, IP address, mobile station identification (MSID) number, or username.
- **Monitor:** Monitor logging records all activity associated with a particular session. This functionality is available in order to comply with law enforcement agency requirements for monitoring capabilities of particular subscribers. Monitors can be performed based on a subscriber's MSID or username.
- **Crash:** Crash logging stores useful information pertaining to system software crashes. This information is useful in determining the cause of the crash.

Configuring Event Logging Parameters

The system can be configured to generate logs based on user-defined filters. The filters specify the facilities (system tasks or protocols) that the system is to monitor and severity levels at which to trigger the generation of the event entries.

Event logs are stored in system memory and can be viewed via the CLI. There are two memory buffers that store event logging information. The first buffer stores the active log information. The second buffer stores inactive logging information. The inactive buffer is used as a temporary repository to allow you to view logs without having data be overwritten. Logs are copied to the inactive buffer only through manual intervention.

Each buffer can store up to 50,000 events. Once these buffers reach their capacity, the oldest information is removed to make room for the newest.

To prevent the loss of log data, the system can be configured to transmit logs to a syslog server over a network interface.

Configuring Event Log Filters

Follow the example below to configure run time event logging parameters for the system:

```
configure
logging filter runtime facility <facility> level <report_level>
logging display { event-verbosity | pdu-data | pdu-verbosity }
end
```

Notes:

- Configure the logging filter that determines which system facilities should be logged and at what levels. Refer to [Specifying Facilities](#) and [Event Severities](#) for detailed information.
- Repeat for every facility that you would like to log.
- *Optional:* Configure event ID restrictions by adding the **logging disable eventid** command. The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Repeat to disable logging for additional event IDs or event ID ranges.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring syslog Servers

Information generated by the run time event logging filters can be transmitted to a syslog server for permanent storage.

 **Important:** The data transmitted to the syslog server is meant to be used for informational purposes. Functions such as billing and performance monitoring should not be based on syslogs.

 **Important:** Although the system provides the flexibility to configure syslog servers on a context-by-context basis, it is recommended that all servers be configured in the *local* context in order to isolate the log traffic from the network traffic.

Use the following example to configure syslog servers:

```
configure
  context local
    logging syslog <ip_address>
  end
```

Notes:

- A number of keyword options/variables are available for the **logging syslog** command. Refer to *Context Configuration Mode Commands* in the *Command Line Interface Reference* for more information.
- Repeat as needed to configure additional syslog servers. There is no limit to the number of syslog servers that can be configured.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring Active Logs

Active logs are event logs that are operator configurable on a CLI instance-by-CLI instance basis. Active logs configured by an administrative user in one CLI instance are not displayed to an administrative user in a different CLI instance. Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as they are generated.

Active logs are not written to the active memory buffer by default. To write active logs to the active memory buffer execute the following command in the Global Configuration mode:

```
[local]<hostname>(config)# logging runtime buffer store all-events
```

When active logs are written to the active memory buffer, they are available to all users in all CLI instances.

Use the following example to configure active logging in Global Configuration mode:

```
[local]<hostname>(config)# logging filter runtime facility facility level  
report_level
```

Notes:

- Configure the logging filter that determines which system facilities should be logged and at what levels. Refer to [Specifying Facilities](#) and [Event Severities](#) for detailed information.
- Repeat for every facility that you would like to log.
- *Optional:* Configure event ID restrictions by adding the **logging disable eventid** command. The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Repeat to disable logging for additional event IDs or event ID ranges.
- A number of keyword options/variables are available for the Exec mode **logging active** command. Refer to *Exec Mode Commands (D-S)* in the *Command Line Interface Reference* for more information.

Once all of the necessary information has been gathered, the Active log display can be stopped by entering the following command in the Exec mode:

```
no logging active
```

Specifying Facilities



Important: The actual facilities available for logging vary by platform type (ASR 5000 or ASR 5500), StarOS version and installed product licenses.

The following facilities can be configured for logging event data:

- **a10:** A10 interface facility
- **a11:** A11 interface facility
- **a11mgr:** A11 Manager facility
- **aaa-client:** Authentication, Authorization and Accounting (AAA) client facility
- **aaamgr:** AAA manager logging facility
- **aaaproxy:** AAA Proxy facility
- **aal2:** ATM Adaptation Layer 2 (AAL2) protocol logging facility
- **acl-log:** Access Control List (ACL) logging facility
- **acsctrl:** Active Charging Service (ACS) Controller facility
- **acsmgr:** ACS Manager facility
- **afctrl:** Fabric Controller facility [ASR 5500 only]
- **afmgr:** Fabric Manager logging facility [ASR 5500 only]
- **alarmctrl:** Alarm Controller facility
- **alcap:** Access Link Control Application Part (ALCAP) protocol logging facility
- **alcapmgr:** ALCAP manager logging facility
- **all:** All facilities
- **asngwmgr:** Access Service Network (ASN) Gateway Manager facility
- **asnpcmgr:** ASN Paging Controller Manager facility
- **bfd:** Bidirectional Forwarding Detection (BFD) protocol logging facility
- **bgp:** Border Gateway Protocol (BGP) facility
- **bindmux:** IPCF BindMux-Demux Manager logging facility
- **bngmgr:** Broadband Network Gateway (BNG) Demux Manager logging facility
- **bssap+:** Base Station Sub-system Application Part+ protocol facility for the login interface between the SGSN and the MSC/VLR (2.5G and 3G)
- **bssgp:** Base Station Sub-system GPRS Protocol logging facility handles exchange information between the SGSN and the BSS (2.5G only)
- **callhome:** Call Home application logging facility
- **cap:** CAMEL Application Part (CAP) logging facility for protocol used in prepaid applications (2.5G and 3G)
- **cbsmgr:** Cell Broadcasting Service (CBS) logging facility [HNBGW]
- **cdf:** Charging Data Function (CDF) logging facility
- **cgw:** Converged Access Gateway (CGW) logging facility
- **cli:** Command Line Interface (CLI) logging facility

- **connproxy**: Controller Proxy logging facility
- **credit-control**: Credit Control (CC) facility
- **cscf**: IMS/MMD Call Session Control Function (CSCF)
- **cscfcpmgr**: CSCFCPMGR logging facility
- **cscfmgr**: SIP CSCF Manager facility
- **cscfnpdb**: CSCF Number Portability Database (NPDB) logging facility
- **cscfrtcp**: IMS/MMD CSCF RTCP log facility
- **cscfrtp**: IMS/MMD CSCF RTP log facility
- **cscftmgr**: SIP CSCF Tunnel and Transport Manager facility
- **esp**: Card/Slot/Port controller facility
- **css**: Content Service Selection (CSS) facility
- **css-sig**: CSS RADIUS Signaling facility
- **cx-diameter**: Cx Diameter Messages facility
- **dcardctrl**: IPsec Daughter Card Controller logging facility
- **dcardmgr**: IPsec Daughter Card Manager logging facility
- **demuxmgr**: Demux Manager API facility
- **dgmbmgr**: Diameter Gmb Application Manager logging facility
- **dhcp**: Dynamic Host Configuration Protocol (DHCP) logging facility
- **dhcpxv6**: DHCPv6
- **dhost**: Distributed Host logging facility
- **diabase**: Diabase messages facility
- **diameter**: Diameter endpoint logging facility
- **diameter-acct**: Diameter Accounting
- **diameter-auth**: Diameter Authentication
- **diameter-dns**: Diameter DNS subsystem
- **diameter-ecs**: ACS Diameter signaling facility
- **diameter-hdd**: Diameter Horizontal Directional Drilling (HDD) Interface facility
- **diameter-svc**: Diameter Service
- **diamproxy**: DiamProxy logging facility
- **dpath**: IPsec Data Path facility
- **drvctrl**: Driver Controller facility
- **eap-ipsec**: Extensible Authentication Protocol (EAP) IPsec facility
- **eap-sta-s6a-s13-s6b-diameter**: EAP/STA/S6A/S13/S6B Diameter messages facility
- **ecs-css**: ACSMGR <-> Session Manager Signalling Interface facility
- **egtpc**: eGTP-C logging facility
- **egtpmgr**: enhanced GPRS Tunneling Protocol (eGTP) manager logging facility
- **egtpu**: eGTP-U logging facility

- **epdg**: evolved Packet Data (ePDG) gateway logging facility
- **event-notif**: Event Notification Interface logging facility
- **evlog**: Event log facility
- **famgr**: Foreign Agent manager logging facility
- **firewall**: Firewall logging facility
- **fng**: Femto Network Gateway (FNG) logging facility
- **gmm**:
 - For 2.5G: Logs the GPRS Mobility Management (GMM) layer (above LLC layer)
 - For 3G: Logs the access application layer (above the RANAP layer)
- **gprs-app**: GPRS Application logging facility
- **gprs-ns**: GPRS Network Service Protocol (layer between SGSN and the BSS) logging facility
- **gq-rx-tx-diameter**: Gq/Rx/Tx Diameter messages facility
- **gss-gcdr**: GTP Storage Server GCDR facility
- **gtpc**: GTP-C protocol logging facility
- **gtpcmgr**: GTP-C protocol manager logging facility
- **gtp**: GTP-prime protocol logging facility
- **gtpu**: GTP-U protocol logging facility
- **gtpumgr**: GTP-U Demux manager
- **gx-ty-diameter**: Gx/Ty Diameter messages facility
- **gy-diameter**: Gy Diameter messages facility
- **hamgr**: Home Agent manager logging facility
- **hat**: High Availability Task (HAT) process facility
- **hdctrl**: HD Controller logging facility
- **henbapp**: Home Evolved NodeB (HENB) App facility
- **henbgw**: HENB-Gateway facility
- **henbgwdemux**: HENB-GW Demux facility
- **henbgwmgr**: HENB-GW Manager facility
- **hnb-gw**: HNB-GW (3G Femto GW) logging facility
- **hnbmgr**: HNB-GW Demux Manager logging facility
- **hss-peer-service**: Home Subscriber Server (HSS) Peer Service facility
- **igmp**: Internet Group Management Protocol (IGMP)
- **ikev2**: Internet Key Exchange version 2 (IKEv2)
- **ims-authorizatn**: IP Multimedia Subsystem (IMS) Authorization Service facility
- **ims-sh**: HSS Diameter Sh Interface Service facility
- **imsimgr**: SGSN IMSI Manager facility
- **imsue**: IMS User Equipment (IMSUE) facility
- **ip-arp**: IP Address Resolution Protocol facility

- **ip-interface**: IP interface facility
- **ip-route**: IP route facility
- **ipms**: Intelligent Packet Monitoring System (IPMS) logging facility
- **ipne**: IP Network Enabler (IPNE) facility
- **ipsec**: IP Security logging facility
- **ipsg**: IP Service Gateway interface logging facility
- **ipsgmgr**: IP Services Gateway facility
- **ipsp**: IP Pool Sharing Protocol logging facility
- **kvstore**: Key/Value Store (KVSTORE) Store facility
- **l2tp-control**: Layer 2 Tunneling Protocol (L2TP) control logging facility
- **l2tp-data**: L2TP data logging facility
- **l2tpdemux**: L2TP Demux Manager logging facility
- **l2tpmgr**: L2TP Manager logging facility
- **lagmgr**: Link Aggregation Group (LAG) manager logging facility
- **lcs**: Location Services (LCS) logging facility
- **ldap**: Lightweight Directory Access Protocol (LDAP) messages logging facility
- **li**: Refer to the *Lawful Intercept Interface Reference* for a description of this command.
- **linkmgr**: SGSN/BSS SS7 Link Manager logging facility (2.5G only)
- **llc**: Logical Link Control (LLC) Protocol logging facility; for SGSN: logs the LLC layer between the GMM and the BSSGP layers for logical links between the MS and the SGSN
- **local-policy**: Local Policy Service facility
- **location-service**: Location Services facility
- **m3ua**: M3UA Protocol logging facility
- **magmgr**: Mobile Access Gateway manager logging facility
- **map**: Mobile Application Part (MAP) protocol logging facility
- **megadiammgr**: MegaDiameter Manager (SLF Service) logging facility
- **mme-app**: Mobility Management Entity (MME) Application logging facility
- **mme-misc**: MME miscellaneous logging facility
- **mmedemux**: MME Demux Manager logging facility
- **mmemgr**: MME Manager facility
- **mmgr**: Master Manager logging facility
- **mobile-ip**: Mobile IP processes
- **mobile-ip-data**: Mobile IP data facility
- **mobile-ipv6**: Mobile IPv6 logging facility
- **mpls**: Multiprotocol Label Switching (MPLS) protocol logging facility
- **mrme**: Multi Radio Mobility Entity (MRME) logging facility
- **mseg-app**: Mobile Services Edge Gateway (MSEG) application logging facility

- **mseg-gtpc**: MSEG GTP-C application logging facility
- **mseg-gtpu**: MSEG GTP-U application logging facility
- **msegmgr**: MSEG Demux Manager logging facility
- **mtp2**: Message Transfer Part 2 (MTP2) Service logging facility
- **mtp3**: Message Transfer Part 3 (MTP3) Protocol logging facility
- **multicast-proxy**: Multicast Proxy logging facility
- **nas**: Non-Access Stratum (NAS) protocol logging facility [MME 4G]
- **npuctrl**: Network Processor Unit Control facility
- **npudrv**: Network Processor Unit Driver facility [ASR 5500 only]
- **npumgr**: Network Processor Unit Manager facility
- **npumgr-acl**: NPUMGR ACL logging facility
- **npumgr-drv**: NPUMGR DRV logging facility
- **npumgr-flow**: NPUMGR FLOW logging facility
- **npumgr-fwd**: NPUMGR FWD logging facility
- **npumgr-init**: NPUMGR INIT logging facility
- **npumgr-lc**: NPUMGR LC logging facility
- **npumgr-port**: NPUMGR PORT logging facility
- **npumgr-recovery**: NPUMGR RECOVERY logging facility
- **npumgr-vpn**: NPUMGR VPN logging facility
- **npusim**: NPUSIM logging facility [ASR 5500 only]
- **ntfy-intf**: Notification Interface logging facility [Release 12.0 and earlier versions only]
- **ogw-app**: Offload Gateway (OGW) application logging facility [Release 12.0 and earlier versions only]
- **ogw-gtpc**: OGW GTP-C application logging facility [Release 12.0 and earlier versions only]
- **ogw-gtpu**: OGW GTP-U application logging facility [Release 12.0 and earlier versions only]
- **ogwmgr**: OGW Demux Manager logging facility [Release 12.0 and earlier versions only]
- **orbs**: Object Request Broker System logging facility
- **ospf**: OSPF protocol logging facility
- **ospfv3**: OSPFv3 protocol logging facility
- **p2p**: Peer-to-Peer Detection logging facility
- **pccmgr**: Intelligent Policy Control Function (IPCF) Policy Charging and Control (PCC) Manager library
- **pdg**: Packet Data Gateway (PDG) logging facility
- **pdgdmgr**: PDG Demux Manager logging facility
- **pdif**: Packet Data Interworking Function (PDIF) logging facility
- **pgw**: Packet Data Network Gateway (PGW) logging facility
- **phs**: Payload Header Suppression (PHS)
- **phs-control**: PHS X1/X5 and X2/X6 Interface logging facility
- **phs-data**: PHS Data logging facility

- **phs-eapol**: PHS EAP over LAN (EAPOL) logging facility
- **phsgwmgr**: PHS Gateway Manager facility
- **phspcmgr**: PHS Paging Controller Manager facility
- **pmm-app**: Packet Mobility Management (PMM) application logging facility
- **ppp**: Point-To-Point Protocol (PPP) link and packet facilities
- **pppoe**: PPP over Ethernet logging facility
- **push**: VPNMGR CDR push logging facility
- **radius-acct**: RADIUS accounting logging facility
- **radius-auth**: RADIUS authentication logging facility
- **radius-coa**: RADIUS change of authorization and radius disconnect
- **ranap**: Radio Access Network Application Part (RANAP) Protocol facility logging info flow between SGSN and RNS (3G)
- **rct**: Recovery Control Task logging facility
- **rdt**: Redirect Task logging facility
- **resmgr**: Resource Manager logging facility
- **rf-diameter**: Diameter Rf interface messages facility
- **rip**: Routing Information Protocol (RIP) logging facility [RIP is not supported at this time.]
- **rohc**: Robust Header Compression (RoHC) facility
- **rsvp**: Reservation Protocol logging facility
- **rua**: RANAP User Adaptation (RUA) [3G Femto GW - RUA messages] logging facility
- **s1ap**: S1 Application Protocol (S1AP) Protocol logging facility
- **sabp**: Service Area Broadcast Protocol (SABP) logging facility
- **saegw**: System Architecture Evolution (SAE) Gateway facility
- **sccp**: Signalling Connection Control Part (SCCP) Protocol logging (connection-oriented messages between RANAP and TCAP layers).
- **sct**: Shared Configuration Task logging facility
- **sctp**: Stream Control Transmission Protocol (SCTP) Protocol logging facility
- **sess-gr**: SM GR facility
- **sessctrl**: Session Controller logging facility
- **sessmgr**: Session Manager logging facility
- **sesstrc**: session trace logging facility
- **sft**: Switch Fabric Task logging facility
- **sgs**: SGs interface protocol logging facility
- **sgsn-app**: SGSN-APP logging various SGSN “glue” interfaces (for example, between PMM, MAP, GPRS-FSM, SMS).
- **sgsn-failures**: SGSN call failures (attach/activate rejects) logging facility (2.5G)
- **sgsn-gtpc**: SGSN GTP-C Protocol logging control messages between the SGSN and the GGSN
- **sgsn-gtpu**: SGSN GTP-U Protocol logging user data messages between the SGSN and GGSN

- **sgsn-mbms-bearer**: SGSN Multimedia Broadcast/Multicast Service (MBMS) Bearer app (SMGR) logging facility
- **sgsn-misc**: Used by stack manager to log binding and removing between layers
- **sgsn-system**: SGSN System Components logging facility (used infrequently)
- **sgsn-test**: SGSN Tests logging facility; used infrequently
- **sgtpcmgr**: SGSN GTP-C Manager logging information exchange through SGTPC and the GGSN
- **sgw**: Serving Gateway facility
- **sh-diameter**: Sh Diameter messages facility
- **sitmain**: System Initialization Task main logging facility
- **sm-app**: SM Protocol logging facility
- **sms**: Short Message Service (SMS) logging messages between the MS and the SMSC
- **sndcp**: Sub Network Dependent Convergence Protocol (SNDCP) logging facility
- **snmp**: SNMP logging facility
- **sprmgr**: IPCF Subscriber Policy Register (SPR) manager logging facility
- **srdp**: Static Rating Database
- **srp**: Service Redundancy Protocol (SRP) logging facility
- **sscfnni**: SSCFNNI Protocol logging facility
- **sscop**: SSCOP Protocol logging facility
- **ssh-ipsec**: SSH IP Security logging facility
- **ssl**: Secure Socket Layer (SSL) message logging facility
- **stat**: Statistics logging facility
- **supserv**: Supplementary Services logging facility [H.323]
- **system**: System logging facility
- **tacacsplus**: TACACS+ Protocol logging facility
- **tcap**: TCAP Protocol logging facility
- **testctrl**: Test Controller logging facility
- **testmgr**: Test Manager logging facility
- **threshold**: threshold logging facility
- **ttg**: Tunnel Termination Gateway (TTG) logging facility
- **tucl**: TCP/UDP Convergence Layer (TUCL) logging facility
- **udr**: User Data Record (UDR) facility (used with the Charging Service)
- **user-data**: User data logging facility
- **user-l3tunnel**: User Layer 3 tunnel logging facility
- **usertcp-stack**: User TCP Stack
- **vpn**: Virtual Private Network logging facility
- **wimax-data**: WiMAX DATA
- **wimax-r6**: WiMAX R6

Configuring Trace Logging

Trace logging is useful for quickly resolving issues for specific sessions that are currently active. They are temporary filters that are generated based on a qualifier that is independent of the global event log filter configured using the **logging filter** command in the Exec mode. Like event logs, however, the information generated by the logs is stored in the active memory buffer.

All debug level events associated with the selected call are stored.



Important: Trace logs impact session processing. They should be implemented for debug purposes only.

Use the following example to configure trace logs in the Exec mode:

```
[local]<hostname>#logging trace { callid call_id | ipaddr ip_address | msid ms_id  
| username username }
```

Once all of the necessary information has been gathered, the trace log can be deleted by entering the following command:

```
[local]<hostname>#no logging trace { callid call_id | ipaddr ip_address | msid  
ms_id | username username }
```

Configuring Monitor Logs

Monitor logging records all activity associated with all of a particular subscriber's sessions. This functionality is available in compliance with law enforcement agency requirements for monitoring capabilities of particular subscribers.

Monitors can be performed based on a subscriber's MSID or username, and are only intended to be used for finite periods of time as dictated by the law enforcement agency. Therefore, they should be terminated immediately after the required monitoring period.

This section provides instructions for enabling and disabling monitor logs.

Enabling Monitor Logs

Use the following example to configure monitor log targets:

```
configure
  logging monitor { ip_addr | IPv6_addr | msid <id> | username <name> }
end
```

Repeat to configure additional monitor log targets.

Disabling Monitor Logs

Use the following example to disable monitor logs:

```
configure
  no logging monitor { ip_addr | IPv6_addr | msid <id> | username <name> }
end
```

Viewing Logging Configuration and Statistics

Logging configuration and statistics can be verified by entering the following command from the Exec mode:

```
[local]<hostname># show logging [ active | verbose ]
```

When no keyword is specified, the global filter configuration is displayed as well as information about any other type of logging that is enabled.

The following table provides information] and descriptions of the statistics that are displayed when the **verbose** keyword is used.

Table 18. Logging Configuration and Statistics Commands

Field	Description
General Logging Statistics	
Total events received	Displays the total number of events generated by the system.
Number of applications receiving events	Displays the number of applications receiving the events.
Logging Source Statistics	
Event sequence ids by process	Displays a list of system processes that have generated events and the reference identification number of the event that was generated.
Msg backlog stat with total cnt	Displays the number of event messages that have been back logged in comparison to the total number of events generated.
LS L2 filter drop rate	Displays the percentage of logging source (LS) layer 2 (L2) event drops.
Abnormal Log Source Statistics	Displays abnormal logging source (LS) statistics, if any.
Runtime Logging Buffer Statistics	
Active buffer	Displays the number of events currently logged in the active memory buffer as well as a date/time timestamp for the oldest and most recent entries in the buffer.
Inactive buffer	Displays the number of events currently logged in the inactive memory buffer.

Viewing Event Logs Using the CLI

Event logs generated by the system can be viewed in one of the following ways:

- **From the syslog server:** If the system is configured to send logs to a syslog server, the logs can be viewed directly on the syslog server.
- **From the system CLI:** Logs stored in the system memory buffers can be viewed directly from the CLI.
- **From the console port:** By default, the system automatically displays events over the console interface to a terminal provided that there is no CLI session active.

This section provides instructions for viewing event logs using the CLI. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 Copy the active log memory buffer to the inactive log memory buffer.

When the active log memory buffer is copied to the inactive log memory buffer existing information in the inactive log memory buffer is deleted.

Both active and inactive event log memory buffers can be viewed using the CLI in Exec mode. However, it is preferable to view the inactive log in order to prevent any data from being over-written. The information from the active log buffer can be copied to the inactive log buffer by entering the following command:

```
[local]<hostname># logs checkpoint
```

Step 2 View the logs by entering the following command:

```
[local]<hostname># show logs
```



Important: A number of optional keywords/variables are available for the **show logs** command. Refer to *Exec Mode Show Commands (H-L)* in the *Command Line Interface Reference* for more information.

Configuring and Viewing Crash Logs

In the unlikely even of a software crash, the system stores information that could be useful in determining the reason for the crash. This information can be maintained in system memory or it can be transferred and stored on a network server.

The system supports the generation of the following two types of logs:

- **Crash log:** Crash logs record all possible information pertaining to a software crash (full core dump). Due to their size, they can not be stored in system memory. Therefore, these logs are only generated if the system is configured with a Universal Resource Locator (URL) pointing to a local device or a network server where the log can be stored.
- **Abridged crash log:** Crash event records are automatically generated when a software crash occurs and are stored in flash memory on management cards. The abridged crash log contains a list crash event records along with associated dump files. This log allows you to view event records and dump files via CLI commands.

Configuring Software Crash Log Destinations

The system can be configured to store software crash log information to any of the following locations:

- On the ASR 5000:
 - **CompactFlash™:** Installed on the SMC [abridged crash log and associated dump files only]
 - **PCMCIA Flash Card:** Installed in the PCMCIA1 slot on the SMC
- On the ASR 5500
 - **Flash memory:** Installed on the active MIO/UMIO [abridged crash log and associated dump files only]
 - **USB memory stick:** Installed in the USB slot on the active MIO/UMIO
- **Network Server:** Any workstation or server on the network that the system can access using the Trivial File Transfer Protocol (TFTP), the File Transfer Protocol (FTP), the Secure File Transfer Protocol (SFTP), or the Hyper-Text Transfer Protocol (HTTP); this is recommended for large network deployments in which multiple systems require the same configuration

Crash log files (full core dumps) are written with unique names as they occur to the specified location. The name format is *crash-card-cpu-time-core*. Where *card* is the card slot, *cpu* is the number of the CPU on the card, and *time* is the Portable Operating System Interface (POSIX) timestamp in hexadecimal notation.

Use the following example to configure a software crash log destination in the Global Configuration mode:

```
configure
  crash enable [ encrypted ] url <crash_url>
end
```

Notes:

- Refer to *Global Configuration Mode Commands* in the *Command Line Interface Reference* for more information on this command.
- Repeat to configure additional software crash log destinations. There is no limit to the number of destinations that can be configured.

Save the configuration as described in *Verifying and Saving Your Configuration*.

Viewing Abridged Crash Log Information Using the CLI

You can view abridged crash information that is stored as a set of event records in flash memory on management cards (**/flash/crashlog2**). Each crash event record has an associated dump file (minicore, NPU or kernel) that can also be displayed (**/flash/crsh2**)

Follow the instructions in this section to view software crash events that have occurred on the system. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 View a list of software crash events by entering the following Exec mode command:

 **Important:** The resulting output may not be the same for all platforms:

```
[local]<hostname># show crash
```

A sample output for **show crash list** is displayed below.

```
==          =====          =====          =====          =====          =====
#           Time           Process      Card/CPU/      SW           HW_SER_NUM
           Time           Process      PID            VERSION      Card
==          =====          =====          =====          =====          =====
1  2012-Aug-01+14:11:15  npumgr      01/1/05414  12.3(44720)  NA
2  2012-Aug-01+14:11:16  npumg       02/1/05430  12.3(44720)  NA
```

The following table provides descriptions for the individual columns displayed in the output.

Column Title	Description
#	Displays an internal reference number for this event record in the log.
Time	Indicates the date and time that the software crash occurred.
Process	Indicates the software task that experienced the crash.
Card	Indicates the card on which the software task was running.
CPU	Indicates the CPU on which the software task was running.
PID	Indicates the process identification (PID) number of the software task that experienced the crash.
SW_Version	Indicates the version of software that experienced the crash.
HW_SER_NUM Card	Indicates the hardware serial numbers of the card where the crash occurred.

Step 2 View the dump file associated with a specific crash event by entering the following Exec mode command:

```
[local]<hostname># show crash
```

crash_number is the number of the crash event as displayed in the output of the **show crash list** command. The information contained in the dump file helps identify and diagnose any internal or external factors causing the software to crash. The following is a sample of the output.

```

***** CRASH #30 *****
Build: 4.0(5800)
Fatal Signal 11: Segmentation fault
PC: [ 0x484650c] strlen()
Faulty address: (nil)
Signal detail: address not mapped to object
Recent events (oldest first):
  [ 0x38f0498] xtcp_client_timer_tick()
  [ 0x391c630] xtcp_wagg_tick()
  [ 0x391c630] xtcp_wagg_tick()
  [ 0x391c630] xtcp_wagg_tick()
  [ 0x2c77cb0] snreactor_file_cb()
  [ 0x2c77cb0] snreactor_file_cb()
  [ 0x3932868] sn_epoll_run_events()
  [ 0x3932868] sn_epoll_run_events()
  [ 0x2c77cb0] snreactor_file_cb()
  [ 0x3932868] sn_epoll_run_events()
Process: card=8 cpu=0 pid=917 argv0=orbs
Crash time: 2011-Dec-23+12:53:19
Recent errno: 11 Resource temporarily unavailable
Registers:
   zr      at      v0      v1      a0      a1      a2      a3
00000000 109b20c4 00000000 00000000 00000000 00000000 01010101 80808080
   t0      t1      t2      t3      t4      t5      t6      t7
00002050 109dbb1c 00000040 00000007 00000000 2abbe9b0 00000000 00000001
   s0      s1      s2      s3      s4      s5      s6      s7
00000000 7fff6c58 7fff6f38 7fff74a8 7fff74a8 7fff6fc8 7fff7168 00000001
   t8      t9      k0      k1      gp      sp      s8      ra
00000000 048464b0 000000f6 00000000 10c1f9f0 7fff6bc0 7fff72d8 048eca80
Stack: 5192 bytes dumped starting from 0x7fff6850
  [ 0x484650c] strlen() sp=0x7fff6bc0
  [ 0x48eca80] __cxa_bad_typeid() sp=0x7fff6be0
  [0x7fff6f40] <trampoline/gdb/stack>() sp=0x7fff6c10
*****

```

Step 3 View the dump files associated with all crash events in the crash log by entering the following Exec mode command:

```
[local]<hostname># show crash
```

Saving Log Files

Log files can be saved to a file in a local or remote location specified by a URL. Use the following Exec mode command to save log files:

```
save logs { url } [active] ] [inactive] [callid call_id] [event-verbosity  
evt_verbosity] [facility facility] [level severity_level] [pdu-data pdu_format]  
[pdu-verbosity pdu_verbosity] [since from_date_time[until to_date_time] ] [ | {  
grep grep_options | more } ]
```

For detailed information on the **save logs** command, see *Exec Mode Commands* in the *Command Line Interface Reference*.

Event ID Overview

 **Important:** The use of event IDs depends on the platform type and the licenses running on the platform.

Identification numbers (IDs) are used to reference events as they occur when logging is enabled on the system. As described previously, logs are collected on a per facility basis. Each facility possesses its own range of event IDs as indicated in the following table.

Table 19. System Facilities and Event ID Ranges

Facility	Description	Event ID Range
a10	A10 Protocol Facility	28000-28999
a11	A11 Protocol Facility	29000-29999
a11mgr	A11 Manager Facility	9000-9999
aaa-client	AAA Client Facility	6000-6999
aaamgr	AAA Manager Facility	36000-36999
aaaproxy	AAA Proxy Facility	64000-64999
aal2	AAL2 Protocol Facility	173200-173299
acl-log	IP Access Control List (ACL) Facility	21000-21999
acsctrl	Active Charging Service Controller (ACSCtrl) Facility	90000-90999
acsmgr	Active Charging Service Manager (ACSMgr) Facility	91000-91999
afctrl	Ares Fabric Controller (ASR 5500 only)	186000-186999
afmgr	Ares Fabric Manager (ASR 5500 only)	187000-187999
alarmctrl	Alarm Controller Facility	65000-65999
alcap	Access Link Control Application Part (ALCAP) Protocol Facility	160900-161399
alcapmgr	ALCAP Manager Facility	160500-160899
asf	ASF Facility	73000-73999
asfprt	ASFPRT Facility	59000-59999
asnswmgr	Access Service Network (ASN) Gateway Manager Facility	100000-100499
asnpcmgr	ASN Paging/Location-Registry Manager Facility	100500-100999
bcmcs	Broadcast/Multicast Service (BCMCS) Facility	109000-109999
bfd	Bidirectional Forwarding Detection (BFD) Protocol Facility	170500-170999
bgp	Border Gateway Protocol (BGP) Facility	85000-85999
bindmux	BindMux Manager Facility [Intelligent Policy Control Function (IPCF)]	158200-158999

Facility	Description	Event ID Range
bngmgr	Broadband Network Gateway (BNG) Manager Facility	182000-182999
bssap	Base Station System Application Part+ (BSSAP+) Service Facilities	131000-131199
bssgp	Base Station System GPRS Protocol (BSSGP) Facility	115050-115099
callhome	Call Home Facility	173600-173999
cap	CAMEL Application Part (CAP) Facility	87900-88099
chatconf	CHATCONF Facility	74000-74999
cli	Command Line Interface (CLI) Facility	30000-30999
connproxy	Connection Proxy Facility	190000-190999
crdt-ctl	Credit Control Facility	127000-127999
cscf	Call Session Control Function (CSCF) Facility	105000-108924
cscfcpmgr	CSCF CP Manager Facility	197000-197999
cscfmgr	CSCF FM Manager Facility	101000-101999
cscfnpdb	CSCF NPDB Facility	108925-108949
cscfrtcp	CSCF RTCP Facility	108976-108999
cscfrtp	CSCF RTP Facility	108950-108975
cscftmgr	CSCF TT Manager Facility	163000-163499
csg	Closed Subscriber Groups (CSG) Facility	188000-188999
csg-acl	CSG Access Control List (ACL) Facility	189000-189999
csp	Card/Slot/Port (CSP) Facility	7000-7999
css	Content Steering Service (CSS) Facility [ESC]	77000-77499
css-sig	Content Service Selection (CSS) RADIUS Signaling Facility	77500-77599
cx-diameter	Cx Diameter Message Facility	92840-92849
dcardctrl	Daughter Card Controller Facility	62000-62999
dcardmgr	Daughter Card Manager Facility	57000-57999
demuxmgr	Demux Manager Facility	110000-110999
dgmbmgr	Diameter Gmb (DGMB) Application Manager Facility	126000-126999
dhcp	DHCP Facility	53000-53999
dhcpv6	DHCPv6 Protocol Facility	123000-123999
dhost	Distributed Host Manager Facility	83000-83999
diameter	Diameter Endpoint Facility	92000-92599
diabase	Diabase Message Facility	92800-92809
diameter-acct	Diameter Accounting Protocol Facility	112000-112999

Facility	Description	Event ID Range
diameter-auth	Diameter Authentication Protocol Facility	111000-111999
diameter-dns	Diameter DNS Subsystem Facility	92600-92699
diameter-ecs	ECS Diameter Signaling Facility	81990-81999
diameter-hdd	Diameter Horizontal Directional Drilling (HDD) Interface Facility	92700-92799
diameter-svc	Diameter Service Facility	121200-121999
diamproxy	Diameter Proxy Facility	119000-119999
dpath	Data Path for IPsec Facility	54000-54999
drvctrl	Driver Controller Facility	39000-39999
ds3mgr	DS3 and DS3/E Line Card Manager Facility (part of NPU Manager Controller Facility)	40000-40999
eap-diameter	Extensible Authentication Protocol (EAP) Diameter Facility	92870-92879
eap-ipsec	EAP IPsec Facility	118000-118999
ecs-css	ACS Session Manager (ACSMgr) Signalling Interface Facility	97000-97099
edr	Event Data Record (EDR) Facility	80000-80999
egtpc	eGTP-C Facility	141000-141999
egtpmgr	eGTP Manager Facility	143000-143999
egtpu	eGTP-U Facility	142000-142999
epdg	Evolved Packet Data Gateway (ePDG) Facility	178000-178999
evlog	Event Log Facility	2000-2999
famgr	Foreign Agent (FA) Manager Facility	33000-33999
firewall	Firewall Facility	96000-96999
fng	Femto Network Gateway (FNG) Facility	149000-149999
gcdr	GGSN-Charging Data Record (G-CDR) Facility	66000-66999
gmm	GPRS Mobility Management (GMM) Facility	88100-88299
gprs-app	General Packet Radio Service (GPRS) Application Facility	115100-115399
gprs-ns	GPRS-NS Protocol Facility	115000-115049
gq-rx-tx-diameter	Gq/Rx/Tx Diameter Messages Facility	92830-92839
gss-gcdr	GTPP Storage Server GCDR Facility	98000-98099
gtpc	GTPC Protocol Facility	47000-47999
gtpcmgr	GTPC Signaling Demultiplexer Manager Facility	46000-46999
gtp	GTP-PRIME Protocol Facility	52000-52999
gtpu	GTPU Protocol Facility	45000-45999
gtpumgr	GTPU Manager Facility	157200-157999

Facility	Description	Event ID Range
gx-ty-diameter	Gx/Ty Diameter Messages Facility	92820-92829
gy-diameter	Gy Diameter Messages Facility	92810-92819
h248prt	H.248 Protocol Facility	42000-42999
hamgr	Home Agent (HA) Manager Facility	34000-34999
hat	High Availability Task (HAT) Facility	3000-3999
hdctrl	Hard Disk (HD) Controller Facility	132000-132999
hddshare	HDD Share Facility	184000-184999
henb-gw	Home eNodeB-GW Facility	195000-195999
henbapp	Home eNodeB Application Facility	196000-196999
henbgwdemux	Home eNodeB-GW Demux Facility	194000-194999
henbgwmgr	Home eNodeB-GW Manager Facility	193000, 193999
hnb-gw	Home NodeB (HNB) Gateway Facility	151000-151999
hnbmgr	HNB Manager Facility	158000-158199
hss-peer-service	Home Subscriber Server (HSS) Facility [MME]	138000-138999
igmp	Internet Group Management Protocol (IGMP) Facility	113000-113999
ikev2	IKEv2 Facility	122000-122999
ims-authorization	IMS Authorization Service Library Facility	98100-98999
ims-sh	IMS SH Library Facility	124000-124999
imsimgr	International Mobile Subscriber Identity (IMSI) Manager Facility	114000-114999
imsue	IMS User Equipment (IMSUE) Facility	144000-145999
ip-arp	IP Address Resolution Protocol (ARP) Facility	19000-19999
ip-interface	IP Interface Facility	18000-18999
ip-route	IP Route Facility	20000-20999
ipms	Intelligent Packet Monitoring System (IPMS) Facility	134000-134999
ipne	IP Network Enabler (IPNE) Facility	192000-192999
ipsec	IPSec Protocol Facility	55000-56998
ipsg	IP Services Gateway (IPSG) Facility	128000-128999
ipsgmgr	IPSG Manager (IPSGMgr) Facility	99000-99999
ipsp	IP Pool Sharing Protocol (IPSP) Facility	68000-68999
kvstore	Key/Value Store (KVSTORE) Facility	125000-125999
l2tp-control	L2TP Control PDU Protocol Facility	50000-50999
l2tp-data	L2TP Data PDU Protocol Facility	49000-49999

Facility	Description	Event ID Range
l2tpdemux	L2TP Demux Facility	63000-63999
l2tpmgr	L2TP Manager Facility	48000-48999
lagmgr	Link Aggregation Group (LAG) Manager Facility	179000-179999
ldap	Lightweight Directory Access Protocol (LDAP) Request Facility	160000-160499
li	Lawful Intercept (LI) Log Facility	69000-69999
linkmgr	Link Manager Facility	89500-89999
llc	Logical Link-Control (LLC) Layer Facility (GPRS)	115700-115799
local-policy	Local Policy Configuration Facility	161400-162399
m3ua	MTP Level 3 (M3UA) Protocol Facility [SIGTRAN]	87500-87699
magmgr	Mobile Access Gateway (MAG) Manager Facility	137500-137999
map	Mobile Application Part (MAP) Protocol Facility [SS7]	87100-87299
megadiammgr	MegaDiameter Manager Facility	121000-121199
mme-app	Mobility Management Entity (MME) Application Facility	147000-147999
mme-misc	MME Miscellaneous Facility	155800-156199
mmedemux	MME Demux Manager Facility	154000-154999
mmemgr	MME Manager Facility	137000-137499
mmgr	Master Manager (MMGR) Facility	86000-86399
mobile-ip	Mobile IP (MIP) Protocol Facility	26000-26999
mobile-ip-data	MIP Tunneled Data Facility	27000-27999
mobile-ipv6	Mobile IPv6 Facility	129000-129999
mpls	Multiprotocol Label Switching (MPLS) Facility	163500-163999
mseg-app	Mobile Services Edge Gateway (MSEG) Application Facility Not supported in this release.	172300-172999
mseg-gtpc	MSEG GTPC Application Facility Not supported in this release.	172000-172199
mseg-gtpu	MSEG GTPU Application Facility Not supported in this release.	172200-172299
msegmgr	MSEG Manager Facility Not supported in this release.	171000-171999
mtp2	Message Transfer Part 2 (MTP2) Service Facility [SS7]	116900-116999
mtp3	Message Transfer Part 3 (MTP3) Service Facility [SS7]	115600-115699
multicast-proxy	Multicast Proxy Facility	94000-94999
nas	Network Access Signaling (NAS) Facility	153000-153999

Facility	Description	Event ID Range
netwstrg	Network Storage Facility	78000-78999
npuctrl	Network Processing Unit (NPU) Control Facility	16000-16999
npudrv	NPU Driver Facility	191000-191999
npumgr	NPU Manager (NPUMGR) Facility	17000-17999
npumgr-acl	NPUMGR ACL Facility	169000-169999
npumgr-drv	NPUMGR Driver Facility	185000-185999
npumgr-flow	NPUMGR Flow Facility	167000-167999
npumgr-fwd	NPUMGR Forwarding Facility	168000-168999
npumgr-init	NPUMGR Initialization Facility	164000-164999
npumgr-lc	NPUMGR LC Facility	180000-180999
npumgr-port	NPUMGR Port Facility	166000-166999
npumgr-recovery	NPUMGR Recovery Facility	165000-165999
npumgr-vpn	NPUMGR VPN Facility	181000-181999
npusim	NPUSIM Facility	176000-176999
ntfy-intf	Event Notification Interface Facility	170000-170499
orbs	Object Request Broker (ORB) System Facility	15000-15999
ospf	Open Shortest Path First (OSPF) Protocol Facility	38000-38999
ospfv3	OSPFv3 Protocol Facility [IPv6]	150000-150999
p2p	Peer-to-Peer (P2P) Facility	146000-146999
pccmgr	Policy Charging and Control (PCC) Manager Facility	159000-159499
pdg	Packet Data Gateway (PDG) Facility	152010-152999
pdgdmgr	PDG TCP Demux Manager (pdgdmgr) Facility (this is a customer-specific facility)	162400-162999
pdif	Packet Data Interworking Function (PDIF) Facility	120000-120999
pgw	Packet Data Network Gateway (PGW) Facility	139000-139999
phs	Payload Header Suppression (PHS) Facility	177000-177999
phs-control	PHS X1/X5 and X2/X6 Interface Facility	136000-136899
phs-data	PHS Data Facility	136900-136979
phs-eapol	PHS EAP over LAN (EAPOL) Facility	136980-136999
phsgwmgr	PHS Gateway Manager Facility	135000-135499
phspcmgr	PHS Paging Controller Manager Facility	135500-135999
pmm-app	Packet Mobility Management (PMM) Application Facility [SGSN]	89200-89499
ppp	Point-To-Point Protocol (PPP) Facility	25000-25999

Facility	Description	Event ID Range
pppoe	Point-to-Point Protocol over Ethernet (PPPoE) Facility	183000-183999
ptt	PTT Facility	76000-76999
push	PUSH (VPNMgr CDR Push) Facility	133000-133999
radius-acct	RADIUS Accounting Protocol Facility	24000-24999
radius-auth	RADIUS Authentication Protocol Facility	23000-23999
radius-coa	RADIUS Change of Authorization (CoA) and Disconnect Facility	70000-70999
ranap	Radio Access Network Application Part (RANAP) Facility	87700-87899
rct	Recovery Control Task (RCT) Facility	13000-13999
rdt	Redirector Task (RDT) Facility	67000-67999
resmgr	Resource Manager (RM) Facility	14000-14999
rf-diameter	Rf Diameter Messages Facility	92860-92869
rip	Routing Information Protocol (RIP) Facility	35000-35999
rohc	Robust Header Compression (ROHC) Protocol Facility	103000-103999
rsvp	RSVP Protocol Facility	93000-93999
rua	RANAP User Adaptation (RUA) Protocol Facility	152000-152009
slap	S1 Application Protocol (S1AP) Facility	155200-155799
saegw	System Architecture Evolution Gateway Facility	191000-191999
sccp	Signalling Connection Control Part (SCCP) Protocol Facility [SS7]	86700-86899
sct	Shared Configuration Task (SCT) Facility	32000-32099
sctp	Stream Control Transmission Protocol (SCTP) Protocol Facility	87300-87499
sess-gr	SESS-GR Facility	77600-77999
sessctrl	Session Controller Facility	8000-8999
sessmgr	Session Manager Facility	10000-12999
sesstrc	Session Trace Facility	155000-155199
sft	Switch Fabric Task (SFT) Facility	58000-58999
sgs	SGs Interface Protocol Facility [MME]	173000-173199
sgsn-app	SGSN Application Interface Facility	115900-115999
sgsn-failures	SGSN Call Failures Facility	89100-89199
sgsn-gtpc	SGSN GTP-C Protocol Facility	116000-116599
sgsn-gtpu	SGSN GTP-U Protocol Facility	86900-87099
sgsn-mbms-bearer	SGSN MBMS Bearer Application (SMGR) Facility	116600-116799
sgsn-misc	SGSN Miscellaneous Facility	88800-89099

Facility	Description	Event ID Range
sgsn-system	SGSN System Components Facility	86400-86499
sgsn-test	SGSN Tests Facility	88700-88799
sgsn2	SGSN2 Facility	114000-117999
sgtpcmgr	SGSN GTP-C (SGTPC) Manager Facility	117000-117999
sgw	Serving Gateway (SGW) Facility	140000-140999
sh-diameter	Sh Diameter Messages Facility	92850-92859
sipcdprt	SIPCDPRT Facility	95000-95999
sitmain	System Initiation Task (SIT) Main Facility	4000-4999
sm-app	Short Message Service (SMS) Facility	88300-88499
sms	SMS Service Facility	116800-116899
sndcp	Sub Network Dependent Convergence Protocol (SNDCP) Facility	115800-115899
snmp	Simple Network Management Protocol (SNMP) Facility	22000-22999
sprmgr	Subscriber Policy Register (SPR) Manager Facility	159500-159999
srdp	Static Rating Database Facility	102000-102999
srp	Service Redundancy Protocol (SRP) Facility	84000-84999
sscfnni	SSCFNNI Protocol Facility [ATM]	115500-115599
sscop	SSCOP Protocol Facility [ATM]	115400-115499
ssh-ipsec	SSH IP Security Facility	56999-56999
ssl	SSL Facility (this is a customer-specific facility)	156200-157199
stat	Statistics Facility	31000-31999
system	System Facility	1000-1999
tacacs+	TACACS+ Protocol Facility	37000-37999
taclep	TACLCP Facility	44000-44999
tcap	Transaction Capabilities Application Part (TCAP) Protocol Logging Facility [SS7]	86500-86699
testctrl	Test Controller Facility	174000-174999
testmgr	Test Manager Facility	175000-175999
threshold	Threshold Facility	61000-61999
ttg	Tunnel Termination Gateway (TTG) Facility	130000-130999
tucl	TCP/UDP Convergence Layer (TUCL) Facility [SS7]	88500-88699
udr	User Data Record (UDR) Facility	79000-79999
user-data	User-Data Facility	51000-51999
user-l3tunnel	User L3 Tunnel Facility	75000-75999

Facility	Description	Event ID Range
usertcp-stack	User TCP Stack Facility	173300-173499
vim	Voice Instant Message (VIM) Facility	60000, 60999
vinfo	VINFO Facility	82000, 82999
vmgctrl	Virtual Media Gateway (VMG) Controller Facility	41000, 41999
vmgctxmgr	VMG Context Manager Facility	43000, 43999
vpn	Virtual Private Network (VPN) Facility	5000-5999
wimax-data	WiMAX DATA Facility	104900-104999
wimax-r6	WiMAX R6 Protocol (Signaling) Facility	104000-104899

Event Severities

The system provides the flexibility to configure the level of information that is displayed when logging is enabled. The following levels are supported:

- **critical:** Logs only those events indicating a serious error has occurred that is causing the system or a system component to cease functioning. This is the highest severity level.
- **error:** Logs events that indicate an error has occurred that is causing the system or a system component to operate in a degraded state. This level also logs events with a higher severity level.
- **warning:** Logs events that may indicate a potential problem. This level also logs events with a higher severity level.
- **unusual:** Logs events that are very unusual and may need to be investigated. This level also logs events with a higher severity level.
- **info:** Logs informational events and events with a higher severity level.
- **trace:** Logs events useful for tracing and events with a higher severity level.
- **debug:** Logs all events regardless of the severity.

Each of the above levels correspond to the “severity” level of the event ID. Therefore, only those event IDs with a “severity” level equal to the logging level are displayed.

Understanding Event ID Information in Logged Output

This section explains the event information that is displayed when logging is enabled.

The following displays a sample output for an event that was logged.

```
2011-Dec-11+5:18:41.993 [cli 30005 info] [8/0/609 <cli:8000609>
_commands_cli.c:1290] [software internal system] CLI session ended for Security
Administrator admin on device /dev/pts/2
```

The following table describes the elements of contained in the sample output.

Table 20. Event Element Descriptions

Element	Description
2011-Dec-11+5:18:41.993	Date/Timestamp indicating when the event was generated
[cli 30005 info]	Information about the event including: <ul style="list-style-type: none"> • The facility the event belongs to • The event ID • The event's severity level In this example, the event belongs to the CLI facility, has an ID of 3005, and a severity level of "info".
[8/0/609 <cli:8000609> _commands_cli.c:1290]	Information about the specific CLI instance.
[software internal system]	Indicates that the event was generated because of system operation.
CLI session ended for Security Administrator admin on device /dev/pts/2	The event's details. Event details may, or may not include variables that are specific to the occurrence of the event.

Chapter 12

Troubleshooting

This segment provides information and instructions for using the system command line interface (CLI) for troubleshooting any issues that may arise during system operation.

Refer to the *ASR 5500 Installation Guide* for comprehensive descriptions of the hardware components addressed by these troubleshooting procedures.

The following topics are included:

- [Detecting Faulty Hardware](#)
- [Taking Corrective Action](#)
- [Verifying Network Connectivity](#)
- [Using the System Diagnostic Utilities](#)

Detecting Faulty Hardware

When power is applied to the chassis, power is sequentially applied to the Management I/O (MIO) cards, Management I/O Universal (UMIO) cards, Data Processing Cards (DPCs), Data Processing Universal Cards (UDPCs), Fabric and Storage Cards (FSCs) and System Status Cards (SSCs).

Each PFU and card installed in the system incorporates light emitting diodes (LEDs) that indicate its operating status. This section describes how to use these status LEDs to verify that all of the installed components are functioning properly.

 **Important:** As the system progresses through its boot process, some cards will not exhibit immediate LED activity. Allow several minutes to elapse after a reboot is initiated before checking the LEDs on the various cards to verify that the boot process has successfully completed.

Licensing Issues

The system boot process is governed by StarOS licenses. During the startup process, each card performs a series of Power-On Self Tests (POSTs) to ensure that the hardware is operational. These tests also verify that the card meets all license requirements to operate in this chassis.

Refer to *Chassis, UMIO and UDPC License Requirements* in the *ASR 5500 Installation Guide* for additional information on the effect licenses and card types have on the boot process.

Using the CLI to View Status LEDs

The status of application and line card LEDs can be viewed through the CLI by entering the **show leds all** command in Exec mode.

The following displays a sample of this command's output.

```

Slot 01: Run/Fail: Green      | Active: Off      | Redundant: Green
Slot 02: Run/Fail: Green      | Active: Off      | Redundant: Green
Slot 03: Run/Fail: Green      | Active: Off      | Redundant: Green
Slot 05: Run/Fail: Green      | Active: Green    | Redundant: Green
      Master: Green
Slot 06: Run/Fail: Green      | Active: Off      | Redundant: Green
      Master: Off
Slot 08: Run/Fail: Green      | Active: Off      | Redundant: Green
Slot 11: Run/Fail: Green      | Active: Green    | Redundant: Green
      Status: Green          | Service: Off
Slot 12: Run/Fail: Green      | Active: Green    | Redundant: Green
      Status: Green          | Service: Off
Slot 13: Run/Fail: Green      | Active: Green    | Redundant: Green
Slot 14: Run/Fail: Green      | Active: Green    | Redundant: Green
Slot 15: Run/Fail: Green      | Active: Green    | Redundant: Green
Slot 16: Run/Fail: Green      | Active: Green    | Redundant: Green
Slot 17: Run/Fail: Green      | Active: Green    | Redundant: Green

```

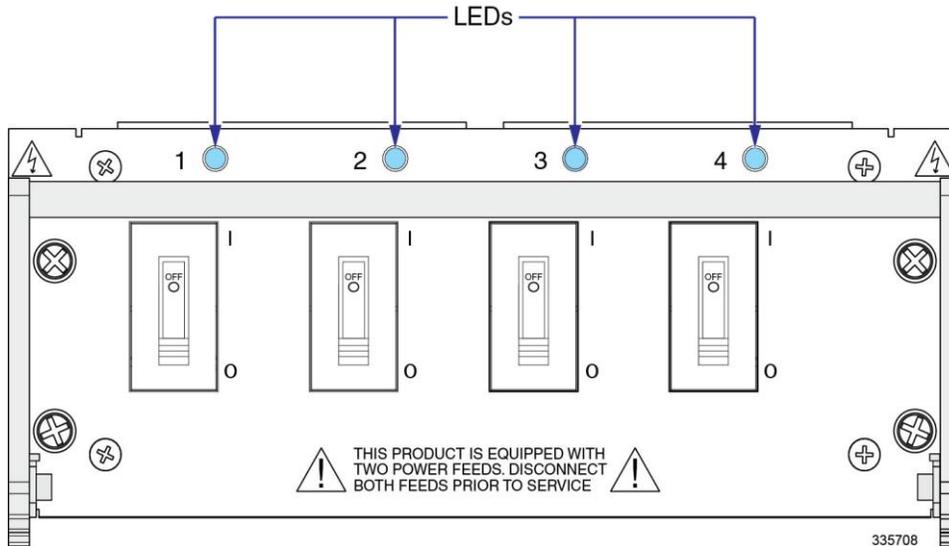
The status of the two Power Filter Units (PFUs) can be viewed by entering the **show power chassis** command in the Exec mode.

Checking the LEDs on the PFU

Each PFU has four LEDs along the top edge of its front panel. You must unsnap the top front cover from the chassis to view these LEDs. Each LED is associated with one of the four -48 VDC power feeds connected to the PFU.

Each LED on the PFU should illuminate blue for normal operating conditions.

Figure 9. PFU LEDs



The possible states for these LEDs are described in the following table. If the LED is not blue, use the troubleshooting information below to diagnose the problem.

Table 21. PFU LED States

Color	Description	Troubleshooting
Blue	Power feed is supplying -48VDC to this power plane	None needed.
None	PFU is not receiving power to one or more of its power planes.	Verify that each circuit breaker is in the ON position. Verify that the RTN and -48VDC lugs are attached properly to the posts on the upper rear of the chassis. Verify that the ground lug is attached properly. Use a voltmeter to verify that the power distribution panel is supplying the correct voltage and sufficient current to the terminals at the rear of the PFU. Check the cables from the power source to the rack for continuity. If a power distribution panel (PDP) is installed between the power distribution frame (PDF) and the chassis, verify that its circuit breakers are set to ON. If a PDP is installed between the PDF and the chassis, check the cables from the PDP to the chassis for continuity.

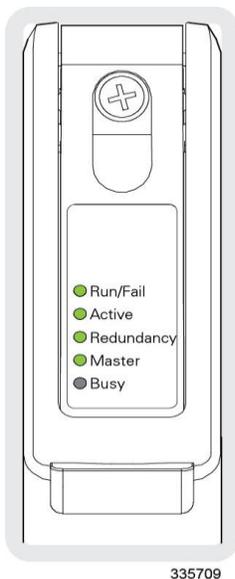
Color	Description	Troubleshooting
		If all of the above suggestions have been verified, then it is likely that the PFU is not functional. Please contact your service representative.

Checking the LEDs on the MIO and UMIO

Each MIO/UMIO is equipped with the following LEDs:

- Run/Fail
- Active
- Redundancy
- Master
- Busy

Figure 10. MIO/UMIO Status LEDs



The possible states for all MIO/UMIO LEDs are described in the sections that follow.

MIO/UMIO Run/Fail LED States

The MIO/UMIO *Run/Fail* LED indicates the overall status of the card. This LED should be steady green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 22. MIO/UMIO Run/Fail LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	This is normal operation during boot-up.
Red	Card powered with error(s) detected	Errors were detected during the Power On Self Tests (POSTs). It is likely that the errors were logged to the system's command line interface during boot.
None	Card is not receiving power	Verify that the LEDs on the PFUs are blue. If they are not, refer to the Checking the LEDs on the PFU section for troubleshooting information.
		Verify that the power source is supplying ample voltage and current to the chassis.
		Verify that the card is properly installed per the instructions in the <i>ASR 5500 Installation Guide</i> .
		If all of the above suggestions have been verified, it is possible that the MIO is not functional. Please contact your service representative.

MIO/UMIO Active LED States

The *Active* LED on the MIO/UMIO indicates that the software is loaded on the card and it is ready for operation. For the MIO installed in chassis slot 5, this LED should be green for normal operation. For the MIO installed in slot 6, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 23. Active LED States

Color	Description	Troubleshooting
Green	Card is active	None needed for the MIO/UMIO in slot 5. If green for the MIO/UMIO in slot 6, verify that the MIO/UMIO in slot 5 is installed and licensed properly according to the instructions in the <i>ASR 5500 Installation Guide</i> .
Blinking Green	Tasks or processes being migrated from the active MIO to the standby MIO.	Refer to <i>Monitoring the System</i> for information on determining the status of the MIO/UMIO and system software processes.
None	Card is not receiving power OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the MIO/UMIO Run/Fail LED States section for troubleshooting information.

MIO/UMIO Redundancy LED States

The *Redundancy* LED on the MIO/UMIO indicates that software is loaded on the card, but it is serving as a redundant component. For the MIO/UMIO installed in slot 6, this LED should be green for normal operation. For the MIO/UMIO installed in slot 8, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 24. MIO/UMIO Redundancy LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed. If green for the MIO/UMIOs in slot 5 and slot 6, the cards and ports are fully backed up.
Amber	Card or port on this card is not backed up by other MIO.	Check the status of the other MIO/UMIO. If it has failed or one or more of its ports are no longer active, the system can continue to function but redundancy is compromised.
		Refer to <i>Monitoring the System</i> for information on determining the status of the MIO/UMIO and system software processes.
Blinking Amber	Tasks or processes being migrated from the active MIO to the standby MIO.	Refer to <i>Monitoring the System</i> for information on determining the status of the MIO/UMIO and system software processes.
None	Card is not receiving power. OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the MIO/UMIO Run/Fail LED States section for troubleshooting information on.

MIO/UMIO Master LED States

The *Master* LED on the MIO/UMIO indicates whether the card is in Active or Standby mode.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information also provided to diagnose the problem.

Table 25. MIO/UMIO Master LED States

Color	Description	Troubleshooting
Green	This card is the Active MIO.	None needed.
Blinking Green	Tasks or processes being migrated from the active MIO to the standby MIO.	Refer to <i>Monitoring the System</i> for information on determining the status of the MIO/UMIO and system software processes.
None	This card is the Standby MIO OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the MIO/UMIO Run/Fail LED States section for troubleshooting information.
		Refer to <i>Monitoring the System</i> for information on determining the status of the MIO/UMIO and system software processes.

MIO/UMIO Busy LED States

The *Busy* LED on the MIO/UMIO indicates that the card is accessing the RAID solid state drives on the FSCs.

This LED is off when no file storage activity is occurring.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 26. MIO/UMIO Busy LED States

Color	Description	Troubleshooting
Green	Files are being transferred to or accessed from the RAID configuration on the FSCs.	None required.
None	There is no RAID activity OR RAID configuration is unavailable.	Checking the LEDs on the FSC

MIO/UMIO – Interface Link LED States

The *Link* LED associated with a 1000Base-T (management) or 10 Gigabit Ethernet port on an MIO/UMIO daughter card (subscriber traffic) indicates the status of the network link. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 27. MIO/UMIO – Interface Link LED States

Color	Description	Troubleshooting
Green	Link is up	None needed. NOTE: This LED will not indicate the presence of a network link until the interface parameters are set during the software configuration process.
None	No power to card OR Link is down.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power. If it is off, refer to the MIO/UMIO Run/Fail LED States section for troubleshooting information.
		Verify that the interface is cabled properly.
		Verify that the device on which the interface is located is cabled and powered properly.

MIO/UMIO – Interface Activity LED States

The *Activity* LED associated with a 1000Base-T (management) or 10 Gigabit Ethernet port on an MIO/UMIO daughter card (subscriber traffic) indicates the presence of traffic on the network link. This LED should be green when data is being transmitted or received over the interface.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 28. MIO/UMIO – Interface Activity LED States

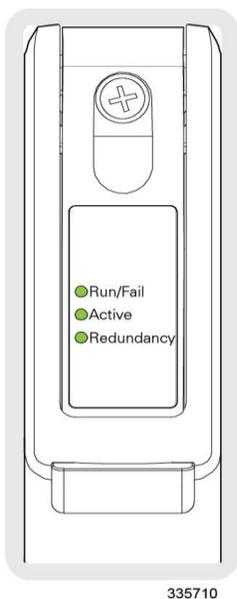
Color	Description	Troubleshooting
Flashing Green	Traffic is present on the link	None needed.
None	No traffic is present on the link	None needed if there is no activity on the link. Prior to interface configuration, this is normal operation.

Checking the LEDs on the DPC and UDPC

Each DPC/UDPC is equipped with status LEDs as listed below:

- Run/Fail
- Active
- Redundancy

Figure 11. DPC/UDPC Status LEDs



The possible states for all of the DPC/UDPC LEDs are described in the sections that follow.

DPC/UDPC Run/Fail LED States

The DPC/UDPC *Run/Fail* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 29. DPC/UDPC Run/Fail LED States

Color	Description	Troubleshooting
Green	Card powered up with no errors detected.	None needed.
Blinking Green	Card is initializing and/or loading software.	This is normal operation during boot-up.
Red	Card powered up with error(s) detected.	Errors were detected during the Power On Self Tests (POSTs). It is likely that the errors were logged to the system's command line interface during boot.
None	Card is not receiving power.	Verify that the LEDs on the PFUs are blue. If they are not, refer to the Checking the LEDs on the PFU section for troubleshooting information.
		Verify that the power source is supplying ample voltage and current to the chassis.
		Verify that the card is properly installed and licensed per the instructions in the <i>ASR 5500 Installation Guide</i> .
		If all of the above suggestions have been verified, it is possible that the DPC/UDPC is not functional. Please contact your service representative.

DPC/UDPC Active LED States

The *Active* LED on the DPC/UDPC indicates that the software is loaded on the card and that the card is ready for operation. When the system first boots up, all installed DPC/UDPCs are booted into standby mode. The system must then be configured as to which DPC/UDPCs should serve as redundant components (remain in standby mode) and which should function as active components.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 30. DPC/UDPC Active LED States

Color	Description	Troubleshooting
Green	Card is active.	The first time power is applied to the system, all of the DPC/UDPCs should be booted into the standby mode. Therefore, this LED should be off.
Blinking Green	Tasks or processes are being migrated from an active DPC to a standby DPC.	Verify that the <i>Redundancy</i> LED on a standby DPC/UDPC is also blinking green. If so, there is an issue with the active DPC/UDPC and it is transferring its processes.
		Refer to <i>Monitoring the System</i> for information on determining the status of the DPC/UDPC and system software processes and functionality.
None	Card is not receiving power. OR Card is in Standby Mode.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the DPC Run/Fail LED States section for troubleshooting information.
		Check the state of the <i>Redundancy</i> LED. If it is green, the card is in standby mode. This is normal operation for the initial power-up. If needed, refer to the <i>Configuring DPC Availability</i> section of <i>System Settings</i> for information on making the card active.

DPC/UDPC Redundancy LED States

The *Redundancy* LED on the DPC/UDPC indicates that software is loaded on the card, but it is serving as a standby component. DPC/UDPCs support n:1 redundancy; the Redundancy LED should be green on only one DPC/UDPC for normal system operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 31. DPC/UDPC Redundancy LED States

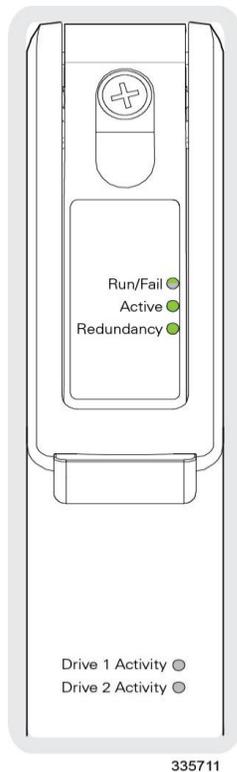
Color	Description	Troubleshooting
Green	Card is in standby mode.	None needed. There is at least one DPC/UDPC in Standby mode.
Amber	Card is not backed up by a standby DPC.	Check the status of the other DPC/UDPCs. If one DPC/UDPC has failed or has been removed from the chassis, the system can continue to function but redundancy is compromised.
		Refer to <i>Monitoring the System</i> for information on determining the status of the DPC/UDPC and system software processes.
Blinking Amber	Tasks or processes being migrated from an active DPC to the standby DPC.	Refer to <i>Monitoring the System</i> for information on determining the status of the DPC/UDPC and system software processes.
None	Card is not receiving power. OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the DPC/UDPC Run/Fail LED States section for troubleshooting information.

Checking the LEDs on the FSC

Each FSC is equipped with the following LEDs as shown in the accompanying figure:

- Run/Fail
- Active
- Redundancy
- Drive 1 Activity
- Drive 2 Activity

Figure 12. FSC Status LEDs



The possible states for all FSC LEDs are described in the sections that follow.

FSC Run/Fail LED States

The FSC *Run/Fail* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 32. FSC Run/Fail LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	This is normal operation during boot-up.
Red	Card powered with error(s) detected	Errors were detected during the Power On Self Tests (POSTs). It is likely that the errors were logged to the system's command line interface during boot.
None	Card is not receiving power	Verify that the LEDs on the PFUs are blue. If they are not, refer to the Checking the LEDs on the PFU section for troubleshooting information.
		Verify that the power source is supplying ample voltage and current to the chassis.
		Verify that the card is properly installed per the instructions in the <i>ASR 5500 Installation Guide</i> .
		If all of the above suggestions have been verified, it is possible that the FSC is not functional. Please contact your service representative.

FSC Active LED States

The *Active* LED on the FSC indicates that the software is loaded on the card and that the card is ready for operation. When the system first boots up, all installed FSCs are booted into ready mode.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 33. FSC Active LED States

Color	Description	Troubleshooting
Green	Card is active.	The first time power is applied to the system, all of the FSCs should be booted into the ready mode. Therefore, this LED should be on.
Blinking Green	Tasks or processes being migrated from an active FSC to a standby FSC.	Verify that the <i>Redundancy</i> LED on a standby FSC is also blinking green. If so, there is an issue with the active FSC and it is transferring its processes.
		Refer to <i>Monitoring the System</i> for information on determining the status of the FSC and system software processes and functionality.
None	Card is not receiving power OR Card in Standby Mode	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the FSC Run/Fail LED States section for troubleshooting information.
		Check the state of the <i>Redundancy</i> LED. If it is green, the card is in standby mode.

FSC Redundancy LED States

The *Redundancy* LED on the FSC indicates that software is loaded on the card, but it is serving as a redundant component. FSC support n+1 redundancy; the Redundancy LED should be green on only one FSC for normal system operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 34. FSC Redundancy LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed. There is at least one FSC in Standby mode.
Amber	Card is not backed up by a standby FSC.	Check the status of the other FSCs. If one FSC has failed or has been removed from the chassis, the system can continue to function but redundancy is compromised.
		Refer to <i>Monitoring the System</i> for information on determining the status of the FSC and system software processes.
Blinking Amber	Tasks or processes being migrated from an active FSC to the standby FSC.	Refer to <i>Monitoring the System</i> for information on determining the status of the FSC and system software processes.
None	Card is not receiving power. OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the FSC Run/Fail LED States section for troubleshooting information.

FSC Drive *n* Activity LED States

The *Drive 1 Activity* and *Drive 2 Activity* LEDs on the FSC indicate that the RAID solid state drives are being accessed by the MIO. Drive 1 and Drive 2 on each FSC form a RAID 0 configuration.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information also provided to diagnose the problem.

Table 35. FSC Drive *n* Activity LED States

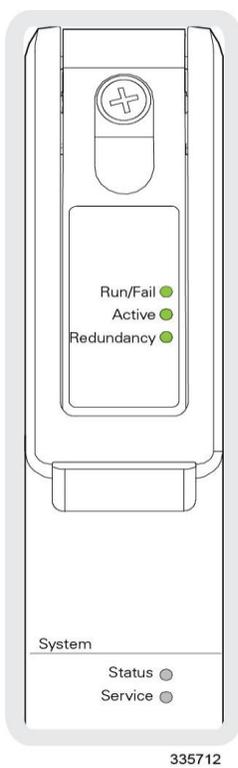
Color	Description	Troubleshooting
Green	Files are being transferred to or accessed from the RAID configuration by the MIO.	None required.
None	There is no RAID activity. OR RAID configuration is unavailable.	Checking the LEDs on the MIO
None	Card is not receiving power	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the FSC Run/Fail LED States section for troubleshooting information.

Checking the LEDs on the SSC

Each SSC is equipped with the following LEDs as shown in the accompanying figure:

- Run/Fail
- Active
- Redundancy
- System Status
- System Service

Figure 13. SSC Status LEDs



The possible states for all SSC LEDs are described in the sections that follow.

SSC Run/Fail LED States

The SSC *Run/Fail* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 36. SSC Run/Fail LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	This is normal operation during boot-up.
Red	Card powered with error(s) detected	Errors were detected during the Power On Self Tests (POSTs). It is likely that the errors were logged to the system's command line interface during boot.
None	Card is not receiving power	Verify that the LEDs on the PFUs are blue. If they are not, refer to the Checking the LEDs on the PFU section for troubleshooting information.
		Verify that the power source is supplying ample voltage and current to the chassis.
		Verify that the card is properly installed per the instructions in the <i>ASR 5500 Installation Guide</i> .
		If all of the above suggestions have been verified, it is possible that the SSC is not functional. Please contact your service representative.

SSC Active LED States

The *Active* LED on the SSC indicates that the software is loaded on the card and that the card is ready for operation. When the system first boots up, both SSCs are booted into ready mode.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 37. SSC Active LED States

Color	Description	Troubleshooting
Green	Card is active.	The first time power is applied to the system, both SSCs should be booted into the ready mode. Therefore, this LED should be on.
Blinking Green	Tasks or processes being migrated from an active FSC to a standby FSC.	Verify that the <i>Redundancy</i> LED on a Istanbul SSC is also blinking green. If so, there is an issue with the active SSC and it is transferring its processes. Refer to <i>Monitoring the System</i> for information on determining the status of the SSC and system software processes and functionality.
None	Card is not receiving power OR Card in Standby Mode	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the <i>SSC Run/Fail LED States</i> section for troubleshooting information.
		Check the state of the <i>Redundancy</i> LED. If it is green, the card is in standby mode.

SSC Redundancy LED States

The *Redundancy* LED on the SSC indicates that software is loaded on the card, but it is serving as a standby component. SSC support 1:1 redundancy; the *Redundancy* LED should be green on the other SSC for normal system operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 38. SSC Redundancy LED States

Color	Description	Troubleshooting
Green	Card is in standby mode	None needed. The other SSC should be in Standby mode.
Amber	Card is not backed up by the standby SSC.	Check the status of the other SSC. If one it has failed or has been removed from the chassis, the system can continue to function but redundancy is compromised.
		Refer to <i>Monitoring the System</i> for information on determining the status of the SSC and system software processes.
Blinking Amber	Tasks or processes being migrated from the active SSC to the standby SSC.	Refer to <i>Monitoring the System</i> for information on determining the status of the SSC and system software processes.
None	Card is not receiving power. OR Card has failed.	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the <i>SSC Run/Fail LED States</i> section for troubleshooting information.

SSC System Status LED States

The *System Status* LED on the SSC indicates the that there is a loss of service somewhere in the system. If this LED is red, the system requires maintenance or service (for example, the system could not locate a a valid software image at boot-up, or a high temperature condition exists).

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information also provided to diagnose the problem.

Table 39. SSC System Status LED States

Color	Description	Troubleshooting
Green	System is operating normally	None required.
Red	The system has experienced a loss of service.	Refer to <i>Monitoring the System</i> for information on determining the status of system hardware and software processes.
None	Card is not receiving power	Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the <i>SSC Run/Fail LED States</i> section for troubleshooting information.

SSC System Service LED States

The *System Service* LED on the SSC illuminates amber to indicate that the system has experienced a hardware component failure.

This LED is off during normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 40. SSC System Service LED States

Color	Description	Troubleshooting
Amber	System requires maintenance (fan filter, temperature warning, PFU outage etc.).	<i>Monitoring the System</i> for show commands, the outputs of which will assist in further determining the problem.
		Refer to <i>System Logs</i> for information on how to view logs.
None	No component failures have been detected. OR Card is not receiving power.	No maintenance needed.

Testing System Alarm Outputs

The system provides the following two physical alarm mechanisms:

- **System Audible Alarm:** Located on the SSC, the speaker is used to provide an audible indicator that a minor, major, or critical alarm has occurred.
- **CO Alarms Interface:** Located on the SSC, this interface provides a DB-15 connector that enables three dry-contact relays (Form C) for the triggering of external audio and/or visual indicators. These indicators can be used to alert that either a minor, major, or critical alarm has occurred.

The operation of these alarms can be tested by issuing the following command:

```
test alarm { audible | central-office [ critical | major | minor ] }
```

Keyword/Variable	Description
audible	Tests the CO Alarm Speaker on the SSC to verify operation.
central-office	Tests the CO Alarm Interface on the MIO to verify operation. Individual alarms can be tested by using one of the following keywords: <ul style="list-style-type: none"> • critical: Specifies that the critical CO Alarms output is to be tested. • major: Specifies that the major CO Alarms output is to be tested. • minor: Specifies that the minor CO Alarms output is to be tested. If no keyword is specified, all alarms are tested.

When this command is executed, the specified alarm is activated for a period of 10 seconds. After this time, the alarm will return to its previous condition.

Taking Corrective Action

In the event that an issue was discovered with an installed application or line card, depending on the severity, it may be necessary to take corrective action.

The system provides several redundancy and fail-over mechanisms to address issues with application and line cards in order to minimize system downtime and data loss. These mechanisms are described in the sections that follow.

Switching MIO/UMIOs

When the system boots up, the MIO/UMIO installed in chassis slot 5 will boot into the Active mode and begin booting other system components. The MIO/UMIO installed in chassis slot 6 will automatically be booted into Standby mode dictating that it will serve as a redundant component. The active MIO/UMIO automatically synchronizes currently running tasks or processes with the standby MIO/UMIO.

In the event of a critical failure on the MIO/UMIO in slot 5, system control will be automatically switched to the standby MIO/UMIO in slot 6. This is a relatively seamless transition because the two are synchronized. The formerly active MIO will then enter the standby mode allowing it to be safely replaced or restored.

In the event that an issue arises that is not severe enough for the system to perform an automatic switchover, a manual switchover can be invoked by executing the following commands from the Exec mode prompt:

```
[local]host_name#
```

Step 1 Initiate a manual MIO/UMIO switch over by entering the following command:

```
card switch from <5 or 6> to <6 or 5>
```

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the switchover.

Step 3 Verify that the switchover was successful by entering the **show card table** command at the Exec mode prompt: Check the entry in the *Oper State* column next to the MIO just switched. Its state should be *Standby*.

Busying Out a DPC/UDPC

This **busy-out** command moves processes from the source DPC/UDPC to the destination DPC/UDPC, or disables the DPC/UDPC from accepting any new calls. When busy-out is enabled, the DPC/UDPC stops receiving new calls but continues to process calls until they are completed. The command prompt is returned once the command is initiated. The busy-out procedure is completed in background.

```
[local]host_name#
```

Step 1 Initiate a busy-out by entering the following command:

```
card busy-out slot_number
```

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

- Step 2** Press **y** to start the switchover.
- Step 3** Verify that the busy-out was successful by entering the **show card table** command at the Exec mode prompt: Check the entry in the *Oper State* column next to the DPC/UDPC just busied-out. Its state should be *Standby*.

Migrating a DPC/UDPC

When the system boots up, all DPC/UDPCs enter the “standby” mode. The standby mode indicates that the card is available for use but is not configured for operation. Installed components can be made active through the software configuration process. Cards that are not configured to enter the “active” mode will remain in standby mode for use as redundant components.

In the event of the critical failure of a DPC/UDPC, tasks will be automatically be migrated from the active card to a redundant card in standby mode.

In the event that an issue arises that is not severe enough for the system to perform an automatic migration, a manual migration can be initiated. Follow the instructions below to manually initiate a DPC/UDPC migration. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

- Step 1** Initiate a DPC/UDPC migration by entering the following command:

```
card migration from original_slot# to final_slot#
```

Keyword/Variable	Description
<i>original_slot#</i>	Specifies the chassis slot number of the DPC/UDPC that is being migrated from, which can be any integer from 1 through 4, and 7 through 10.
<i>final_slot#</i>	Specifies the chassis slot number of the DPC/UDPC that is being migrated to, which can be any integer from 1 through 4, and 7 through 10.

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

- Step 2** Press **y** to start the migration.
- Step 3** Verify that the migration was successful by entering the **show card table** command at the Exec mode prompt. Check the entry in the *Oper State* column next to the packet processing card that was just migrated from. Its state should be *Standby*. The state of the packet processing card migrated to should be *Active*.

Halting Cards

Cards other than MIO/UMIOs that are in either the Active or Standby modes can be halted. Halting these cards places them into the “offline” mode. In this mode, the card is unusable for session processing as either an active or redundant component.

If a card in the active mode is halted, its tasks, processes, or network connections will be migrated or switched to a redundant component prior to entering the offline mode.

This section describes how to initiate a card halt and restore halted components.

Initiate a Card Halt

 **Important:** Do not initiate a **card halt** for an active FSC if there are less than two active FSCs in the system. The system returns an error message if there are less than two active FSCs. There are similar restrictions when executing the **card reboot** or **card upgrade** commands on active FSCs. Refer to the *Command Line Interface Reference* for detailed information.

Follow the instructions below to manually initiate a card halt. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Initiate a manual card migration by entering the following command:

```
card halt slot#
```

slot# is the chassis slot number in which the card to be halted is installed. It can be any integer from 1 through 4, and 7 through 18. You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to initiate the halt operation.

Step 3 Verify that the migration was successful by entering the **show card table** command at the Exec mode prompt.

Check the entry in the *Oper State* column next to the card that was just halted. Its state should be *Offline*. If the card was in active mode prior to the execution of this command, the state of the redundant component associated with it should now be *Active*.

Restore a Previously Halted Card

Follow the instructions below to restore a card that was previously halted. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Reboot the card to be restored by entering the following command.

```
card reboot slot# -force
```

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

- Step 2** Press **y** to start the reboot of the card.
- Step 3** Verify that the migration was successful by entering the **show card table** command at the prompt. Check the entry in the *Oper State* column next to the card that was just restored. Its state should be the state of that it was in before it was halted.

Verifying Network Connectivity

There are multiple commands supported by the system to verify and/or troubleshoot network connectivity. Note that network connectivity can only be tested once system interfaces and ports have been configured and bound.

The commands specified in this section should be issued on a context-by-context basis. Contexts act like virtual private networks (VPNs) that operate independently of other contexts. Ports, interfaces, and routes configured in one context cannot be tested from another context without additional configuration.

To switch between contexts enter the following command at the root prompt for the Exec mode:

```
context context_name
```

context_name is the name of the context to which you wish to switch. The following prompt appears:

```
[context_name]host_name#
```

Using the ping or ping6 Command

The **ping** or **ping6** command verifies the system’s ability to communicate with a remote node in the network by passing data packets between and measuring the response. This command is useful in verifying network routing and if a remote node is able to respond at the IP layer. The command has the following syntax:

Syntax

The **ping** command has the following syntax:

```
ping host_ipv4_address [ count num_packets ] [ flood ] [ pattern packet_pattern ]
[ size octet_count ] [ src { src_host_name | src_host_ipv4_address } ] [ vrf
vrf_name ]

ping6 host_ipv6_address [ count num_packets ] [ flood ] [ pattern packet_pattern ]
[ size octet_count ] [ src { src_host_name | src_host_ipv6_address } ] [ vrf
vrf_name ]
```

Keyword/Variable	Description
<i>host_ipv4_address</i> OR <i>host_ipv6_address</i>	Identifies the remote node to which is the target of the ping command. <i>host_ipv4_address</i> specifies the remote node using its IP address entered in IPv4 dotted-decimal notation. <i>host_ipv6_address</i> specifies the remote node using its IP address entered in IPv6 colon-separated-hexadecimal notation.
count <i>num_packets</i>	Specifies the number of packets to send to the remote host for verification. <i>num_packets</i> is an integer from 1 through 10000. The default is 5.
flood	Sends ping packets as rapidly as possible or 100 per second, whichever is faster. Use with caution. This command keyword terminates after receiving (count) responses. If used against an interface that is not responding, it will run <u>indefinitely</u> .

Keyword/Variable	Description
pattern <i>packet_pattern</i>	Specifies a pattern to use to fill the internet control message protocol packets with. <i>packet_pattern</i> is a hexadecimal value from 0x0000 through 0xFFFF. <i>packet_pattern</i> must begin with “0x” followed by up to four hexadecimal digits. The default is that each octet of the packet is encoded with the octet number of the packet.
size <i>octet_count</i>	Specifies the number of bytes in each IP datagram as an integer from 40 through 18432. The default is 56.
src { <i>src_host_name</i> <i>src_host_ipv4_address</i> OR <i>src_host_ipv6_address</i> }	Specifies an IP address to use in the packets as the source node. <i>src_host_name</i> specifies the source node using the node’s logical host name which must be resolved via DNS lookup. <i>src_host_ipv4_address</i> specifies the source node using its IP address in IPv4 dotted-decimal notation. <i>src_host_ipv6_address</i> specifies the source node using its IP address in IPv6 colon-separated-hexadecimal notation. The default is the IP address of the interface through which the ping was issued.

The following displays a sample of a successful **ping** (IPV4) response.

```

PING 192.168.250.1 (192.168.250.1): 56 data bytes
64 bytes from 192.168.250.1: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.250.1: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=4 ttl=255 time=0.2 ms
--- 192.168.250.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.4 ms
    
```

Troubleshooting

If no response is received from the target follow these troubleshooting procedures:

- Verify that the correct IP address was entered.
- Attempt to ping a different device on the same network. If the ping was successful then it is likely that your system configuration is correct. Verify that the device you are attempting to ping is powered and functioning properly.
- Verify the port is operational.
- Verify that the configuration of the ports and interfaces within the context are correct.
- If the configuration is correct and you have access to the device that you’re attempting to ping, ping the system from that device.
- If there is still no response, it is likely that the packets are getting discarded by a network device. Use the **tracert** or **tracert6** and **show ip static-route** commands discussed in this segment to further troubleshoot the issue.

Using the traceroute or traceroute6 Command

The **traceroute** or **traceroute6** command collects information on the route data will take to a specified host. This is a useful troubleshooting command that can be used to identify the source of significant packet delays or packet loss on the network. This command can also be used to identify bottle necks in the routing of data over the network.

traceroute – IPv4

The **traceroute** command has the following syntax:

```
traceroute { host_name | host_ipv4_address } [ count packets ] [ df ] [ maxttl
max_ttl ] [ minttl min_ttl ] [ port port_number ] [ size octet_count ] [ src {
src_host_name | src_host_ipv4_address } ] [ timeout seconds ] [ vrf vrf_nam ]
```

Keyword/Variable	Description
<i>host_name</i>	Identifies the remote node to trace the route to by the hostname. <i>host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup.
<i>host_ipv4_address</i>	Identifies the remote node to trace the route to by its IP address. <i>host_ipv4_address</i> is expressed in IPv4 dotted-decimal notation.
count	Specifies the number of UDP probe packets to send. The default is 3.
df	Indicates the packets for the tracing of the route should not be fragmented. If a packet would require fragmenting then it is dropped and the ICMP response "Unreachable, Needs Fragmentation" is received.
maxttl <i>max_ttl</i>	Specifies the maximum time to live (TTL) for the route tracing packets. The time to live is the number of hops through the network; it is not a measure of time. <i>max_ttl</i> is an integer from 1 through 255. It is an error if <i>max_ttl</i> is less than <i>min_ttl</i> , whether <i>min_ttl</i> is specified or defaulted. The default maximum TTL is 30 hops.
minttl <i>min_ttl</i>	Specifies the minimum time to live for the route tracing packets. The time to live is the number of hops through the network; it is not a measure of time. <i>min_ttl</i> is an integer from 1 through 255. It is an error if <i>min_ttl</i> is greater than <i>max_ttl</i> , whether <i>max_ttl</i> is specified or defaulted. The default minimum TTL is 1 hop.
port <i>port_number</i>	Specifies a specific port to connect as an integer from 1 through 65535. The default port is 33434.
size	Specifies the number of bytes each packet as an integer from 40 through 32768. The default is 40.

Keyword/Variable	Description
src { <i>src_host_name</i> <i>src_host_ipv4_address</i> }	Specifies an IP address to use in the packets as the source node. <i>src_host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup. <i>src_host_ipv4_address</i> specifies the remote node using its IP address in IPv4 dotted-decimal notation. The default is the IP address of the interface through which the ping was issued.
timeout <i>seconds</i>	Specifies the maximum time to wait (in seconds) for a response from each route tracing packet as an integer from 2 through 100. The default is 5.
vrf <i>vrf_name</i>	Specifies an existing VRF name for which routing information will be displayed as an alphanumeric string of 1 through 63 characters.

The following displays a sample output.

```
tracert to 192.168.250.1 (192.168.250.1), 30 hops max, 40 byte packets
 1 192.168.250.1 (192.168.250.1) 0.446 ms 0.235 ms 0.178 ms
```

tracert6 – IPv6

The **tracert6** command has the following syntax:

```
tracert6 { host_name | host_ipv6_address } [ count packets ] [ maxttl max_ttl ] [ port port_number ] [ size octet_count ] [ src { src_host_name | src_host_ipv6_address } ] [ timeout seconds ] [ vrf vrf_name ]
```

Keyword/Variable	Description
<i>host_name</i>	Identifies the remote node to trace the route to by the hostname. <i>host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup.
<i>host_ipv6_address</i>	Identifies the remote node to trace the route to by its IP address. <i>host_ipv6_address</i> is expressed in IPv6 colon-separated-hexadecimal notation.
count	Specifies the number of UDP probe packets to send. The default is 3.
maxttl <i>max_ttl</i>	Specifies the maximum time to live (TTL) for the route tracing packets. The time to live is the number of hops through the network; it is not a measure of time. <i>max_ttl</i> is an integer from 1 through 255. It is an error if <i>max_ttl</i> is less than <i>min_ttl</i> , whether <i>min_ttl</i> is specified or defaulted. The default maximum TTL is 30 hops.
port <i>port_number</i>	Specifies a specific port to connect as an integer from 1 through 65535. The default port is 33434.
size	Specifies the number of bytes each packet as an integer from 40 through 32768. The default is 40.

Keyword/Variable	Description
src { <i>src_host_name</i> <i>src_host_ipv6_address</i> }	Specifies an IP address to use in the packets as the source node. <i>src_host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup. <i>src_host_ipv6_address</i> specifies the remote node using its IP address in IPv6 colon-separated-hexadecimal notation. The default is the IP address of the interface through which the ping was issued.
timeout <i>seconds</i>	Specifies the maximum time to wait (in seconds) for a response from each route tracing packet as an integer from 2 through 100. The default is 5.
vrf <i>vrf_name</i>	Specifies an existing VRF name for which routing information will be displayed as an alphanumeric string of 1 through 63 characters.

The following displays a sample output.

```
tracert6 to 2001:4A2B::1f3F (2001:4A2B::1f3F), 30 hops max, 40 byte packets
 1 2001:4A2B::1f3F (2001:4A2B::1f3F) 0.446 ms 0.235 ms 0.178 ms
```

Viewing IP Routes

The system provides a mechanism for viewing route information to a specific node or for an entire context. This information can be used to verify network connectivity and to ensure the efficiency of the network connection. The command has the following syntax:

```
show ip route [ route_ip_address ]
show ipv6 route [ route_ipv6_address ] ]
```

Keyword/Variable	Description
<i>route_ip_address</i>	Specifies the IP address of a network node for which route information is displayed. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
<i>route_ipv6_address</i>	Specifies the IP address of a network node for which route information is displayed. The address can be entered in IPv6 colon-separated-hexadecimal notation.

If no keywords are specified, all IP routes within the context's routing table are displayed.

The following displays a sample of this command's output showing a context IPv4 routing table.

```
"*" indicates the Best or Used route.
  Destination          Nexthop          Protocol Prec Cost Interface
*0.0.0.0/0             10.0.4.1         static    0    0 SPIO1
*10.0.4.0/24           0.0.0.0          kernel    0    0 SPIO1
*10.0.4.0/32           0.0.0.0          kernel    0    0 SPIO1
*10.0.4.3/32           0.0.0.0          kernel    0    0 SPIO1
*10.0.4.255/32         0.0.0.0          kernel    0    0 SPIO1
```

Viewing the Address Resolution Protocol Table

The system provides a mechanism for viewing Address Resolution Protocol (ARP) table information to a specific node or for an entire context. This information can be used to verify that when the system sends an ARP packet, it receives valid responses from other network nodes. The command has the following syntax:

```
show ip arp [ arp_ip_address ]
```

arp_ip_address specifies a specific network node for which to display ARP information. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. If this keyword is not specified, all entries within the context's ARP table are displayed.

 **Important:** Restarting the VPN Manager removes all interfaces from the kernel which in turn removes all ARP entries. However, the NPU still retains all of the ARP entries so that there is no traffic disruption. From a user point of view, **show ip arp** is broken since this command gathers information from the kernel and not the NPU.

The following displays a sample of this command's output showing a context's ARP table.

```
Flags codes:
C - Completed, M - Permanent, P - Published, ! - Not answered
T - has requested trailers
Address          Link Type Link Address      Flags  Mask Interface
10.0.4.240      ether    00:05:47:02:20:20 C MIO1
10.0.4.7        ether    00:05:47:02:03:36 C MIO1
10.0.4.1        ether    00:01:30:F2:7F:00 C MIO1
```

Using the System Diagnostic Utilities

The system provides protocol monitor and test utilities that are useful when troubleshooting or verifying configurations. The information generated by these utilities can help identify the root cause of a software or network configuration issue.

This section describes how to use these utilities.



Important: Only an administrator with Operator or higher privilege can run the diagnostic utilities described in this section.

Using the Monitor Utility

For troubleshooting purposes, the system provides a protocol monitoring utility. This tool displays protocol information for a particular subscriber session or for every session being processed.



Caution: The monitor tool may cause session processing delays and/or data loss. Therefore, it should be used only when troubleshooting.

Using the Protocol Monitor

The protocol monitor displays information for every session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions below to invoke and configure the protocol monitoring tool.

- Step 1** Invoke the protocol monitor from the Exec mode by entering the **monitor protocol** command.
An output listing all the currently available protocols, each with an assigned number, is displayed.
- Step 2** Choose the protocol that you wish to monitor by entering the associated number at the *Select:* prompt. A right arrow (>) appears next to the protocol you selected.
- Step 3** Repeat *step 2* as needed to choose multiple protocols.
- Step 4** Press **B** to begin the protocol monitor.

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

Step 5 Enter **y** to proceed with the monitor or **n** to go back to the previous menu.

```

C - Control Events      (ON )
D - Data Events        (ON )
E - EventID Info       (ON )
H - Display ethernet   (ON )
I - Inbound Events     (ON )
O - Outbound Events    (ON )
S - Sender Info        (OFF)
T - Timestamps        (ON )
X - PDU Hexdump        (OFF)
A - PDU Hex/Ascii      (OFF)
+/- Verbosity Level   ( 1)
L - Limit Context      (OFF)
M - Match Newcalls     (ON )
R - RADIUS Dict        (no-override)
G - GTPP Dict          (no-override)
Y - Multi-Call Trace  ((OFF))
(Q)uit,  <ESC> Prev Menu,  <SPACE> Pause,  <ENTER> Re-Display Options
    
```

Step 6 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter associated with that option (C, D, E, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys. The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Step 7 Press the **Enter** key to refresh the screen and begin monitoring. The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the Protocol Monitor for a Specific Subscriber

The protocol monitor can be used to display information for a specific subscriber session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool for a specific subscriber session.

Step 1 To invoke the session-specific protocol monitor from the Exec mode enter the **monitor subscriber** command.

```

monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr | msid | msisdn |
next-call | pcf | peer-fa | peer-lac | sgsn-address | type | username }
    
```

Step 2 Specify the method the monitor should use by entering the appropriate keyword.

Step 3 Select other options and/or enter the appropriate information for the selected keyword.

If no session matching the specified criteria was being processed when the monitor was invoked, a screen of available monitoring options appears.

Step 4 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter or 2-digit number associated with that option (C, D, E, 11, 12, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.



Important: Option **y** for performing multi-call traces is only supported for use with the GGSN.

Step 5 Repeat *step 6* as needed to enable or disable multiple protocols.

Step 6 Press **Enter** to refresh the screen and begin monitoring.

The following displays a portion of a sample of the monitor's output for a subscriber named *user2@aaa*. The default protocols were monitored.

```
-----
Incoming Call:
-----
MSID: 0000012345 Callid: 002dc6c2
Username: user2@aaa SessionType: unknown
Status: Active Service Name: xxx1
Src Context: source Dest Context:
-----

<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)
PPP Tx PDU (9)
PAP 9: Auth-Ack(1), Msg=

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (27)
CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA

INBOUND>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (30)
IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-DNS=0.0.0.0,
Sec-DNS=0.0.0.0

<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)
PPP Tx PDU (26)
IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-DNS=0.0.0.0

INBOUND>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70

INBOUND>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (31)
LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)

INBOUND>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0
```

```
<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87

INBOUND>>>>> 10:02:35:519 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the DHCP Testing Tool

The CLI provides a mechanism for testing network connectivity with and configuration of DHCP servers. This functionality can help determine the accuracy of the system’s DHCP configuration and the server’s response time.

This tool provides a mechanism for obtaining an IP address for one or more DHCP servers with which the system communicates.

 **Important:** This tool must be executed from the context in which the DHCP server(s) are configured.

To execute the DHCP test tool enter the following command within the appropriate context:

```
dhcp test dhcp-service { service_name } [ all | server ip_addr ]
```

Table 41. Sample dhcp test dhcp-service Command Output

Keyword	Description
<i>service_name</i>	Name of DHCP service from which to lease an IP address.
all	Test all DHCP servers in this DHCP service.
server	Followed by IP address of the server under test.

Chapter 13

System Recovery

This segment describes how to recover a system after it has failed to complete a reboot following a power off cycle or interruption of the normal boot sequence following a **reload** command.

This segment includes the following sections:

- [Prerequisites](#)
- [Accessing the boot CLI](#)
- [Booting from a Selected Image](#)

 **Caution:** This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

Prerequisites

Successful recovery from a failed reboot requires that you have access to the system via a console port, and have an uncorrupted copy of the StarOS boot image file stored in flash memory on the management card, or accessible from an external memory device.

Console Access

The boot recovery sequence can only be executed via a terminal connected to the serial console port on the active management card. This connection can be through a terminal server that is accessible via a LAN interface.

The boot recovery sequence can only be viewed via the console port.

Boot Image

The boot recovery command line interface allows you to specify from which boot image you would like to boot the system. If the system failed to reload following a software update, you can initiate a boot from a previously stored image.

The system recovery procedure will prompt you to enter the path name for the location of the StarOS boot image from which the system will boot. By default the boot command will timeout and attempt to reload the highest priority image from flash memory using the default configuration file.

A boot image binary file name uses the following format:

```
production.build_number.asr5000 or .asr5500.bin
```

Refer to the *Configuring the Boot Stack* section in *Software Management Operations* for additional information on boot stack entries and prioritization.

Accessing the boot CLI

To access the boot CLI you must interrupt an in-progress reload (reboot) sequence.

 **Caution:** This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

Initiate a Reboot

A reload can be initiated in one of two ways:

- Power cycle the chassis – Turn the circuit breakers on the power filter units (PFUs) Off (O) and then On (I).
- Execute a **reload** command

```
[local]asr5000# reload -noconfirm
```

The boot sequence displays messages on the terminal as it steps through its processes.

Interrupt the Boot Sequence

When the “Booting priority” message line appears (and not before), press CTRL+C to break out of the boot process as shown in the example below:

```
Booting priority 8
 image : /flash/production.35272.st40.bin
 config: /flash/system.cfg
 Entry at 0x00000000cba45e0
```

Press CTRL+C at this point in the sequence.

A message similar to the following appears after the boot process has been interrupted:

```
*****9/0 Ctrl-C Pressed-----
Failed.
 aborted by user
8/0:boot>
```

Enter CLI Mode

With the boot prompt displayed, enter **cli** to access the boot recovery CLI. The CLI prompt changes as shown below:

```
8/0:boot>cli
8/0:cli>
```

boot Command Syntax

The boot recovery command has the following syntax:

```
boot [ -show | -priority=* | -config=* | -noconfig ] { bootfile_URL }
```

The options for this command include:

- **-show**: displays the current boot configuration
- **-priority=***: selects the desired boot stack priority (*)
- **-config=***: enters the desired configuration filename (*), if not the default file
- **-noconfig**: boots using no configuration file

bootfile_URL is the URL for the location of the StarOS boot image file. It specifies the path and file name of the StarOS .bin file from which the system will be booted.

The URL may refer to a local file (flash) or an external file on a memory device attached to the management card. The URL must be entered in the following format:

```
{ /flash | /pcmcial or usb1 /filename
```

Booting from a Selected Image

You will issue a boot command via the boot CLI to initiate the system recovery process.

Boot Using No Configuration File

This procedure boots the system using the specified boot image without also loading a configuration file. A sample command string appears below:

```
8/0:cli>boot -noconfig /flash/production.41731.asr5x00.bin
```

The boot sequence ends with a prompt to enter the Quick Setup Wizard for creating a configuration file.

```
Launching StarOS
Starting program at 0x000000000100000
Starent Networks ASR5x00 Intelligent Mobile Gateway
SMC (or MIO) is starting up.....
Starting software <version> <build_number>...
No configuration found, press enter to continue.
1. Do you wish to continue with the Quick Setup Wizard[yes/no]:
```

You can exit the Quick Setup Wizard by entering **no** in response to the above prompt. Load a desired configuration file using the Exec mode **configure** command followed by the URL for the configuration file as shown in the example below:

```
[local]asr5x00# configure /flash/system.cfg
```

Boot Using A Specified Configuration File

This procedure boots the system using the specified boot image and configuration file. A sample command string appears below:

```
8/0:cli>boot -config=/flash/system.cfg /flash/production.41731.asr5x00.bin
```

The boot sequence ends with the appearance of the CLI prompt.

```
[local]asr5000#
```

Confirm that the desired configuration has loaded by running the Exec mode **show configuration** command.

Appendix A

Engineering Rules

This segment provides engineering guidelines for configuring the system to meet network deployment requirements.

This segment consists of the following topics:

- [CLI Session Rules](#)
- [ASR 5500 Interface and Port Rules](#)
- [Context Rules](#)
- [Subscriber Rules](#)
- [Service Rules](#)
- [Access Control List \(ACL\) Engineering Rules](#)

CLI Session Rules

Multiple CLI session support is based on the amount of available memory. The Resource Manager reserves enough resources to support a minimum of six CLI sessions at all times. One of the six sessions is further reserved for use exclusively by a CLI session on an MIO/UMIO serial interface.

Additional CLI sessions beyond the pre-reserved limit are permitted if sufficient MIO/UMIO resources are available. If the Resource Manager is unable to reserve resources for a CLI session beyond those that are pre-reserved, users with administrator-privileges are prompted to create the new CLI session, even without reserved resources.

ASR 5500 Interface and Port Rules

The rules discussed in this section pertain to the Ethernet ports used for subscriber traffic on the MIO/UMIO card (ports 10 through 29).

- Give all logical interfaces a unique name to identify the interface from others in the same context. Logical interfaces in different contexts may have the same name.
- A single physical port can support multiple logical interfaces when you configure VLAN tags for that physical port. You can use VLAN tagging to bind a single physical port to multiple logical interfaces that reside in different contexts.
- Assign all logical interfaces a valid IP address and subnet.
 - Give each logical interface within a context a unique IP address(es). Logical interfaces in different contexts can have the same IP address(es).
 - If multi-homing is supported on the network, you can assign all logical interfaces a single primary IP address and up to 16 secondary IP addresses.
- You can configure a logical interface in only one context, but you can configure multiple interfaces (up to 512) in a single context.
- You can apply a maximum of 256 access control list (ACL) rules to a single logical interface.
- All ports are identified by their <slot#>/<port#>.
- Each physical port for subscriber traffic on an MIO/UMIO card may contain up to a maximum of 1,024 VLAN tags.
- A logical interface is limited to using a single VLAN on a single physical port, identified by its <card#/slot#/port#>.

Packet Data Network (PDN) Interface Rules

The following engineering rules apply to the interface to the packet data network (PDN):

- Configure the logical interfaces used to facilitate the PDN interface within the egress context.
- The default is to use a single interface within the egress context to facilitate the PDN interface.
- You can configure multiple interfaces in the egress context by using static routes or dynamic routing protocols.
- You may also configure next-hop default gateways.

Context Rules

- A maximum of 63 contexts may be configured per chassis.
- Interfaces per Context
 - 512 Ethernet+PPP+tunnel interfaces
 - 32 ipv6ip tunnel interfaces
 - 511 GRE tunnels (2,048 GRE tunnels per chassis)
 - 256 loopback interfaces
- IP Addresses and IP Address Pools
 - Up to 2,000 IPv4 address pools can be configured within a single context (regardless of the number of packet processing cards) with a total system limit of 5,000 IPv4 address pools for all contexts.
 - Up to 32 IPv6 pools can be configured within a single context.
 - There is also a limit of 4,000,000 pool addresses and 32,000,000 static addresses that can be configured per context. Therefore, the number of pools depends on how many addresses are being used and how they are subnetted.
 - Each context supports up to 32,000,000 static IP pool addresses. You can configure a maximum total of 96,000,000 static IP pool addresses per chassis. Each static IP pool can contain up to 500,000 addresses.
 - Each context supports up to 16,000,000 dynamic IP pool addresses. You can configure a maximum total of 32,000,000 dynamic IP pool addresses per chassis. Each dynamic IP pool can contain up to 500,000 addresses.



Important: Each address in the pool requires approximately 60 bytes of memory. The amount of memory required, however, depends on a number of factors such as the pool type, and hold-timer usage. Therefore, in order to conserve available memory, you may need to limit the number of pools depending on the number of addresses to be configured and the number of installed application cards.

- The maximum number of simultaneous subscriber sessions is controlled by the installed capacity license for the service(s) supported.
- The maximum number of static address resolution protocol (ARP) entries per context is 128.
- The maximum number of domains per context is 2,048.
- ASN-GW services configured within the same context cannot communicate with each other.
- Routes
 - Up to 1,200 static routes per context (48,000 per chassis).
 - 6,000 pool routes per context (6,000 per chassis)
 - 5,000 pool explicit host routes per context (6,000 per chassis)
 - 64 route maps per context
- BGP
 - 16,000 BGP prefixes can be configured per context (64,000 per chassis)
 - 64 EBGP peers can be configured per context (512 per chassis)

- 16 IBGP peers per context
- 512 BGP/AAA monitors per context in support of Interchassis Session Recovery (ICSR)
- OSPF
 - 200 OSPF neighbors per chassis
 - 10,000 OSPF routes per chassis (64,000 per chassis)
- MPLS
 - 16 label distribution protocol (LDP) sessions per context
 - 8,000 forwarding equivalence class (FEC) entries per context (48, 000 per chassis)
 - Up to 8,000 incoming label map (ILM) entries per context (48, 000 per chassis)
- VRF (GGSN)
 - 250 virtual routing and forwarding (VRF) tables per context (1,024 or 2,048 [release 14.0+] VRFs per chassis)
 - 16,384 IP routes
- 128 AAA servers per context for a default AAA server group. The servers can be configured as accounting, authentication, charging servers, or any combination thereof.
- You can configure up to 800 AAA server groups per context with following limitations:
 - 128 servers per AAA server group (accounting, authentication, charging server, or any combination thereof)
 - 1,600 servers per context in AAA Server group mode (accounting, authentication, charging server, or any combination thereof)
 - 800 NAS-IP address/NAS identifier (one primary and one secondary per server group) per context
- Up to 12 charging gateway functions (CGFs) for GTPP accounting can be configured per context.
- Up to 16 bidirectional forwarding detection (BFD) sessions per context (64 per chassis)



Important: Refer to *Engineering Rules* in your product administration guide for additional information on product-specific operating limits.

Subscriber Rules

The following engineering rules apply to subscribers configured within the system:

- Configure a maximum of 2,048 local subscribers per context.
- You may configure attributes for each local subscriber.
- The system creates a default subscriber default for each context when the context is made. Configure attributes for each default subscriber. If a AAA-based subscriber is missing attributes in the authentication reply message, the default subscriber attributes in the context where the subscriber was authenticated are used.



Important: Default is not used when local authentication (for local subscribers) is performed.

- Configure default subscriber templates on a per AAA realm (domain aliases configured within a context) basis.
- Configure default subscriber templates on a per PDSN, FA, ASN-GW, or HA service.
- For AAA authenticated subscribers, the selection of local subscriber template to use for setting attributes is in the following order:
 - If the username (NAI) matches any local domain name and the domain name has a local subscriber name configured, that local subscriber template is used.
 - If the first case fails, and if the serving service has a default username configured, that subscriber template is used.
 - If the first two cases fail, the default subscriber template in the AAA context is used.

Service Rules

The following engineering rules apply to services configured within the system:

- Configure a maximum of 256 services (regardless of type) per system.



Caution: Large numbers of services greatly increase the complexity of management and may affect overall system performance. Therefore, you should not configure a large number of services unless your application absolutely requires it. Please contact your Cisco service representative for more information.

- The total number of entries per table and per chassis is limited to 256.
- Although you can use service names that are identical to those configured in different contexts on the same system, this is not a good practice. Services with the same name can lead to confusion and difficulty in troubleshooting problems, and make it difficult to understand the output of **show** commands.

Access Control List (ACL) Engineering Rules

The following rules apply to Access Control Lists:

- The maximum number of rules per ACL is 128.
- The maximum number of ACL rules applied per port is 128.
- The maximum number of ACL rules applied per context is 1,024.
- The maximum number of ACL rules per IPsec policy is 1.
- The maximum number of IPsec ACL rules per context is 1,024.
- The maximum number of ACLs you can configure per context is limited by the number of rules allowed within each ACL. If each ACL contained the maximum number of rules (128), the maximum number of ACLs per context is 8 ($128 \times 8 \text{ ACLs} = 1,024 \text{ ACL rules per context}$).
- The maximum number of ACLs applied to an IP access group is 1, whether it is configured for a port or context. Since the maximum number of IP access groups you can apply to an interface or context is 16, the following calculations apply:
 - For each interface/port: 8 rules per ACL multiplied by 16 IP access groups = 128 (the ACL rules limit per port)
 - For each context: 64 rules per ACL multiplied by 16 IP access groups = 1,024 (the ACL rules limit per context)

Appendix B

System Software Tasks

This appendix describes the system and subsystem tasks running under StarOS on an ASR 5500 platform.

It includes the following sections:

- [Overview](#)
- [Primary Task Subsystems](#)
- [Primary Subsystem Controllers and Managers](#)

Overview

For redundancy, scalability and robust call processing, StarOS is divided into a series of tasks that perform specific functions. These tasks communicate with each other as needed to share control and data signals. As a result, system processes can be distributed across multiple tasks thus reducing the overall work-load on any given task and improving system performance. This distributed design provides fault containment that greatly minimizes the impact to processes or sessions due to a failure.

All tasks run in a Common Firmware Environment (CFE) that resides on specialized Central Processing Units (CPUs) on each of the MIO/UMIO and DPC/UDPC cards. The Management Input/Output (MIO) and Management Input/Output Universal (UMIO) cards each have a single CPU that is responsible for running tasks related to system management and control. The Data Processing Cards (DPCs) and Data Processing Cards Universal (UDPCs) contain two CPUs (CPU 0 and CPU 1). These CPUs are responsible for session processing and running the various tasks and processes required to handle mobile data calls. In addition to the CPUs, the MIO/UMIOs and DPC/UDPCs have multiple Network Processor Units (NPUs) for IP forwarding.

The following sections describe the primary tasks that are implemented by the system:

- [Primary Task Subsystems](#)
- [Primary Subsystem Controllers and Managers](#)

Primary Task Subsystems

The individual tasks that run on the CPUs are divided into subsystems. Following is a list of the primary subsystems responsible for call session processing:

- **System Initiation Task (SIT):** This subsystem starts tasks and initializes the system. This includes starting a set of initial tasks at system startup time (static tasks), and starting individual tasks on demand at arbitrary times (dynamic tasks).
- **High Availability Task (HAT):** With the Recovery Control Task (RCT) subsystem, the HAT subsystem maintains the operational state of the system. HAT monitors the various software and hardware components of the system. If there are unusual activities, such as the unexpected termination of another task, the HAT subsystem takes a suitable course of action, such as triggering an event to the RCT subsystem to take corrective action or to report the status. The end result is that there is minimal or no impact to the service.
- **Recovery Control Task (RCT):** This subsystem executes a recovery action for any failure that occurs in the system. The RCT subsystem receives signals from the HAT subsystem (and in some cases from the NPU subsystem) and determines what recovery actions are needed.

The RCT subsystem runs on the active MIO/UMIO and synchronizes the information it contains with the RCT subsystem on the standby MIO/UMIO.

- **Shared Configuration Task (SCT):** This subsystem provides a facility to set, retrieve, and receive notification of system configuration parameters. The SCT is mainly responsible for storing configuration data for the applications that run on the system.

The SCT subsystem runs only on the active MIO/UMIO and synchronizes the information it contains with the SCT subsystem on the standby MIO/UMIO.

- **Resource Management (RM):** This subsystem assigns resources, such as CPU loading and memory, for every system task upon start-up. The RM subsystem monitors resource use to verify that allocations are as specified. RM also monitors all sessions and communicates with the Session Controller to enforce capacity licensing limits.
- **Virtual Private Network (VPN):** This subsystem manages the administrative and operational aspects of all VPN-related entities in the system. The functions performed by the VPN subsystem include:
 - Creating separate VPN contexts
 - Starting the IP services within a VPN context
 - Managing IP pools and subscriber IP addresses, and distributing the IP flow information within a VPN context.

All IP operations within the system are done within specific VPN contexts. In general, packets are not forwarded across different VPN contexts. The only exception currently is the Session subsystem.

- **Network Processing Unit (NPU):** This subsystem is responsible for the following:
 - Using the database to match address and port numbers to destination tasks for fast-path forwarding of dataframes
 - Receiving and transmitting user data frames to/from various physical interfaces
 - IP forwarding decisions (both unicast and multicast)
 - Per-interface packet filtering
 - Traffic management and traffic engineering
 - Passing user data frames to/from DPC/UDPC CPUs

- Modifying, adding, or stripping datalink/network layer headers
- Recalculating checksums
- Maintaining statistics
- Managing external Ethernet interfaces
- **Card/Slot/Port (CSP):** Coordinates the events that occur when any card is inserted, locked, unlocked, removed, shutdown, or migrated. CSP also performs auto-discovery and configures ports on a newly-inserted MIO/UMIO card. It determines how MIO/UMIO cards map to packet processing cards.

The CSP subsystem runs only on the active MIO/UMIO and synchronizes the information it contains with the SCT subsystem on the standby MIO/UMIO. It is started by the SIT subsystem and monitored by the HAT subsystem.

- **Session:** Performs high-touch processing of mobile subscribers' packet-oriented data session flows. High-touch user data processing consists of the following:
 - Payload transformation
 - Filtering and scheduling
 - Statistics collection
 - Policing

Primary Subsystem Controllers and Managers

Many of the primary subsystems are composed of critical tasks—controller tasks called Controllers, and subordinated tasks called Managers. Critical tasks are essential to the system's ability to process calls, such as those in the SIT subsystem.

Controllers serve several purposes:

- They monitor the state of their Managers and allow communication between Managers within the same subsystem.
- They enable inter-subsystem communication since they can communicate with the controllers of other subsystems.
- They mask the distributed nature of the software from the user allowing for ease of management.

Managers manage resources and mappings between resources. In addition, some managers are directly responsible for call processing.

The following section provides information about the composition of the primary subsystems that are composed of critical, controller, and / or manager tasks.

ASR 5x00 Subsystems

The following tables describe managers and tasks performing within the specified subsystems on an ASR 5x00 platform.

Important: Variations regarding how the managers and tasks are distributed based on session recovery (SR) are included in the Card and CPU columns in some tables. Tables without these indicators are applicable to ASR 5x00s with and without session recovery. The ASR 5x00 dynamically distributes processes, tasks, and managers on startup. The following tables list the typical locations but variations can occur depending on available resources.

Table 42. ASR 5x00 System Initiation Subsystem

Task	Description	Card	CPU
SITMAIN	Initiated at system start-up,.	All	All
	Reads and provides startup configuration to other SIT components.		
	Starts SITREAP sub-function.		
	Maintains CPU state information.		
SITPARENT sub-function	Starts MIO/UMIOs in either active or standby mode.	All	All
	Registers tasks with HAT task.		
	Notifies CSP task of CPU startup completion.		
	Brings up packet processing cards in standby mode.		
SITREAP sub-function	Shuts down tasks as required.	All	All

Table 43. ASR 5x00 High Availability Subsystem

Task	Description	Card	CPU
HAT System Controller (HATSYSTEM)	Controls all the HAT sub-function tasks in the system. It is initiated on system start-up.	MIO/UMIOs	0
	Initializes system components (such as the Gigabit Ethernet switches and switch fabric).		
	Monitors system components such as fans for state changes.		
	Triggers actions for redundancy in the event of fault detection.		
	The HAT subsystem on the redundant MIO/UMIO mirrors the HAT subsystem on the active MIO/UMIO.		

Task	Description	Card	CPU
HATCPU	Performs device initialization and control functions because of the CPU's hardware capabilities.	All packet processing cards MIO/UMIOs	0
	Reports the loss of any task on its CPU to HATSYSTEM sub-function.		
	Controls the LEDs on the packet processing cards.		
	Initializes and monitors the dedicated hardware on packet processing cards.		
	Collects CPU monitoring information periodically and reports to the master HATCPU sub-function.	All packet processing cards	ALL
	Reports the loss of any task on its CPU to the master HATCPU sub-function.		
	Performs device initialization and control functions because of the CPU's hardware capabilities.		
	Reports the loss of any task on its CPU to HATSYSTEM sub-function.		
	Controls the LEDs on the MIO/UMIO.		
	Initializes and monitors the dedicated hardware on the MIO/UMIOs.		

Table 44. ASR 5x00 Resource Manager (RM) Subsystem

Task	Description	Card	CPU
Resource Manager Controller (RMCTRL)	Started by the SITPARENT task on system startup, and monitored by the HAT task for a failure, the RMCTRL performs the following functions at startup.	Active MIO/UMIO	0
	Initializes resources such as CPUs and memory.		
	Requests updated card status from the CSP subsystem and updates the system card table.		
	Communicates with all RMMGRs to request their most recent set of resource data.		
Resource Manager Managers (RMMGRs)	Started by the SITPARENT task, and monitored by the HAT tasks for failures, each RMMGR performs the following functions at startup:	All	All
	Initializes the local resource data and local resource scratch space.		
	Communicates with the SIT task on the local CPU to get its entire task table and the resources associated with each task.		
	Gathers current resource utilization for each task.		
	Sends the resource data to the RMCTRL task.		

Table 45. ASR 5x00 Virtual Private Networking (VPN) Subsystem

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
VPN Controller (VPNCTRL)	Created at system start-up.	Active MIO/UMIO	Active MIO/UMIO	0	0
	Initiates the VPN Manager for each context.				
	Informs the Session Controller task when there are additions or changes to contexts.				
	Routes context specific operation information to the appropriate VPN Manager.				
	Performs VPN Manager recovery and saves all VPN related configuration information in the SCT task.				
	Only one Session Controller operates at any time.				
VPN Manager (VPNMGR)	Started by the VPN Controller for each configured context (one is always present for the local context).	Active MIO/UMIO (local context)	Active MIO/UMIO (local context)	0 (all contexts)	0 (local context)
	Performs IP address pool and subscriber IP address management.				
	Performs all context specific operations including but not limited to: UCM services, IP interfaces, the Address Resolution Protocol (ARP), IP address pool management, slow path forwarding, NPU flows, port Access Control Lists (ACLs), and logging.				
	Provides IP interface address information for each context to the Session Controller.				
Border Gateway Protocol (BGP)	Created by the VPN Manager for each context that has enabled the BGP routing protocol (router bgp config-context CLI command).	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
	Responsible for learning and redistributing routing information via the BGP protocol.				
	Maintains the BGP peering connections.				
	Applies any defined BGP routing policy.				
Open Shortest Path First (OSPF)	Created by VPN Manager for each context that has enabled the OSPF routing protocol (router ospf config-context CLI command).	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
	Responsible for learning and redistributing routing information via the OSPF protocol.				
	Maintains the OSPF neighboring relationship.				
	Maintains the LSA database.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	Performs SPF calculations.				
	Applies any defined OSPF routing policy				
Open Shortest Path First (OSPFv3)	Created by VPN Manager for each context that has enabled the OSPF routing protocol (router ospfv3 config-context CLI command)	Same as VPN Mgr.			
	Responsible for learning and redistributing routing information via the OSPFv3 protocol.				
	Maintains the OSPFv3 neighboring relationship.				
	Maintains the LSA database.				
	Performs OSPFv3 SPF calculations.				
	Applies any defined OSPFv3 routing policy.				
Routing Information Protocol (RIP)	Created by VPN Manager for each context that has enabled the RIP routing protocol (router rip config-context CLI command)	Same as VPN Mgr.			
	Responsible for learning and redistributing routing information via the RIP protocol.				
	Maintains the RIP database.				
	Sends periodic RIP update messages.				
	Applies any defined RIP routing policy.				
ZEBOSTM OSPF Message	Created by VPN Manager for each context. It is responsible for maintaining the routing table for the context.	Same as VPN Mgr.			
	Maintains the routing table (RIB and FIB).				
	Performs static routing.				
	Interfaces to the kernel for routing & interface updates.				
	Redistributes routing information to dynamic routing protocols.				
	Calculates nexthop reachability.				

Table 46. ASR 5x00 Network Processing Unit (NPU) Subsystem

Task	Description	Card	CPU
NPU Controller (NPUCTRL)	Created at system start-up.	Active MIO/UMIO	0
	Monitors the state of NPU Managers in the system.		
	Registers to receive notifications when NPU Manager crashes.		
	Controls recovery operation.		
	Provides a centralized location for CLI commands related to NPU Manager state.		
	Only one NPU Controller operates in the system at any time.		
NPU Manager (NPUMGR)	Created for every packet processing card installed and started and it performs the following functions.	MIO/UMIOs and packet processing cards	0, 1
	Provides port configuration services to the CSP task.		
	Provides interface binding and forwarding services to the VPN Manager.		
	Provides flow insertion and removal services to Session Manager and AAA Manager tasks.		
	Provides recovery services to the NPU Controller.		

Table 47. ASR 5x00 Session Subsystem

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Session Controller (SESSCTRL)	Created at system start-up.	Active MIO/UMIO	Active MIO/UMIO	0	0
	Acts as the primary point of contact for the Session Subsystem. Since it is aware of the other subsystems running within the system, the Session Controller acts as a proxy for the other components, or tasks, that make up the subsystem.				
	Starting, configuring, and coordinating the efforts of the Session Processing Subsystem sub-managers.				
	Works with Resource Manager to start new Session Managers when all existing Session Managers exceed their capacity.				
	Receives context information from VPN Managers.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	Distributes IP interface address information to other Session Processing Subsystem sub-managers.				
	Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services.				
	Only one Session Controller operating in the system at any time.				
Session Manager (SESSMGR)	Created by the Session Controller.	All packet processing cards	All packet processing cards except the first	0	0 on all packet processing cards except the first
	Provides a subscriber processing system that supports multiple session types.				
	Multiple Session Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system.				
	A single Session Manager can service sessions from multiple A11 Managers, and from multiple contexts.				
	Processes protocols for A10/A11, GRE, R3, R4, R6, GTPU/GTPC, PPP, and Mobile IP.				
	Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services.				
	Session Managers are paired with AAA Managers.				
A11 Manager (A11MGR)	Created by the Session Controller for each context in which a PDSN service is configured.	Active packet processing cards	First packet processing card	0	Any (see NOTE)
	Receives the R-P sessions from the PCF and distributes them to different Session Manager tasks for load balancing.				
	Maintains a list of current Session Manager tasks to aid in system recovery.				
	The A11 Manager task is also known as the Signaling De-multiplexing task (SDT).				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Authorization, Authentication, and Accounting (AAA) Manager (AAAMGR)	Paired with Session Managers (except the one running on the MIO/UMIO).	Active packet processing cards	All packet processing cards (except first)	All except 0	All
	Performs all AAA protocol operations and functions for subscribers and administrative users within the system.	Active MIO/UMIO (CLI only)	Active MIO/UMIO (CLI only)	0	0
	Acts as a AAA client to AAA servers.				
	Manages GTP Prime (GTPP) messaging with charging gateway functions (CGFs).				
	Multiple AAA Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system.				
	AAA operations for the CLI are done through a AAA Manager running on the active MIO/UMIO				
Charging Detail Record Module (CDRMOD)	Responsible for receiving EDR/UDR records from different ACSMGR instances in the system.	First packet processing card	First packet processing card	0	0
	Responsible for writing the received EDR/UDR records in files using the configured file naming conventions.				
Diameter GMB Application Manager (DGMBMGR)	Provides MBMS feature support for GGSN. It is instantiated when an MBMS policy CLI is configured in the GGSN Service configuration mode. DGMBMGR	Active packet processing cards		0	
	Maintains the MBMS UE and bearer contexts.				
	Handles the GMB interface over a Diameter connection to a BMSC Server for MBMS bearer sessions. DGMBMGR recovers by polling all SMGRs for MBMS session states and recreating the MBMS UE and MBMS bearer context information.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Diameter Proxy (DIAMPROXY)	Created by DIACTRL (which runs as part of VPNCTRL) and the number of DIAMPROXY tasks spawned is based on the configuration to use “multiple” or “single” proxies. In instances that a single proxy is configured, only one DIAMPROXY task is spawned for the entire chassis and runs on demux packet processing cards. When multiple proxies are configured, one DIAMPROXY task is run per packet processing card.	Active packet processing cards (see description)	Active packet processing cards (see description)	All (see description)	All (see description)
	Maintains Diameter base connections to all peers configured in the system.				
	Informs applications about any change in the connection status.				
	Acts as a pass-through to the messages from application to the Diameter server.				
	Just acts as a forwarding agent (does not maintain any queues).				
	A single Diameter proxy is used to service multiple Diameter applications.				
eGTP Egress Manager	Created by Session Controller for each context in which an egtp-service of interface type sgw-egress or MME is configured.	Packet processing card	First packet processing card	All	Any (see NOTE)
	Handles certain EGTP messages from SGW, PGW.				
	Maintains list of current EGTP sessions.				
	Maintains list of current Session Manager tasks which aids in session recovery.				
	Handles GTP Echo messaging.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
eGTP Ingress Manager	Created by Session Controller for each context in which an egtp-service of interface type sgw-ingress or pgw-ingress is configured.	Packet processing card	First packet processing card	All	Any (see NOTE)
	Receives EGTP sessions from MME/S4 SGSN/SGW and distributes them to different Session Manager tasks for load balancing.				
	Maintains list of current EGTP sessions.				
	Maintains list of current Session Manager tasks which aids in session recovery.				
	Handles GTP Echo messaging.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.				
GPRS Tunneling Protocol Control (GTPC) Message Manager (GTPCMGR)	Created by the Session Controller for each context in which a GGSN service is configured.	Active packet processing cards		0	
	Receives the GTP sessions from the SGSN and distributes them to different Session Manager tasks for load balancing.				
	Maintains a list of current Session Manager tasks to aid in system recovery.				
	Verifies validity of GTPC messages.				
	Maintains a list of current GTPC sessions.				
	Handles GTPC Echo messaging to/from SGSN.				
GTP-U Manager (GTPUMGR)	Created by the Session Controller for each context in which a GTPU service is configured.	Active packet processing cards	First packet processing card	0	Any (see NOTE)
	Maintains a list of the GTPU-services available within the context and performs load-balancing (of only Error-Ind) for them.				
	Supports GTPU Echo handling.				
	Provides Path Failure detection on no response for GTPU echo.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	<p>Receives Error-Ind and demuxes it to a particular Session Manager.</p> <p>Serves as the Default GTPU listener. GTPUMGR will process GTPU packets with invalid TEID.</p> <p>The above features are supported for both GTPUv0 and GTPUv1.</p> <p>NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.</p>				
Home Agent (HA) Manager (HAMGR)	<p>Created by the Session Controller for each context in which an HA service is configured, the HA Manager performs the following functions:</p> <p>Receives Mobile IP sessions from the Foreign Agents (FAs) and distributes them to different Session Manager tasks.</p> <p>Maintains a list of current Session Manager tasks that aids in system recovery.</p> <p>Functions as the DemuxMgr – handles all the PMIP signaling packets.</p> <p>HAMgr also functions as the Demuxmgr for MIPv6/MIPv4 HA.</p> <p>NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.</p>	Active packet processing cards	First packet processing card	0	Any (see NOTE)
IMSI Manager for MME (IMSIMgr)	<p>Starts when an MME service configuration is detected. There will be only one instance of this task:</p> <p>Signaling De-multiplexer: Selects which SessMgr to use for new subscriber sessions.</p> <p>IMSI-to-SessMgr resolution: Maintains and reports MME-related demux statistics on events like Attach by IMSI, Attach by GUTI, etc.</p>	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	<p>IMSIMgr can interact with the following tasks in the system:</p> <ul style="list-style-type: none"> • Session Controller • MME Manager • Session Manager <p>NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are started.</p>				
International Mobile Subscriber Identity Manager for SGSN (IMSIMgr)	Started by the Session Controller, the IMSIMgr performs the following functions	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)
	Selects SessMgr, when not done by LinkMgr or SGTPCMgr, for calls sessions based on IMSI/P-TMSI.				
	Load-balances across SessMgrs to select one for assigning subscriber sessions to.				
	Maintains records for all subscribers on the system.				
	Maintains mapping between the IMSI/P-TMSI and SessMgrs.				
NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active deumux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.					
IP Services Gateway Manager (IPSGMGR)	Created by the Session Controller.	Active packet processing cards	—	0	—
	In Server mode, acts as a RADIUS server, and supports Proxy functionality.				
	In Snoop mode supports snooping RADIUS Accounting messages.				
	Load balances requests among different SessMgrs.				
	Activates and deactivates sessions.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Layer 2 Tunneling Protocol Manager (L2TPMGR)	Created by the Session Controller for each context in which a LAC or LNS service is configured (additional Managers created as needed depending on loading).	Active packet processing cards	First packet processing card	0	Any (see NOTE)
	Responsible for all aspects of L2TP processing.				
	Maintains protocol state machines for all L2TP sessions and tunnels.				
	Triggers IPSec encryption for new L2TP tunnels as needed.				
	Works with Session Managers to gracefully bring down tunnels.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.				
L2TP Demultiplexor Task (L2TPDEMUX)	Created by the Session Controller when an LNS service is created, only one L2TPDemux task is invoked for the entire system.	Active packet processing cards	First packet processing card.	0	Any (see NOTE)
	De-multiplexes and forwards new incoming tunnel create requests to L2TPMgrs.				
	Maintains information about current active tunnels in all L2TPMgrs.				
	Load balances requests among L2TPMgrs.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.				
Mobile Access Gateway Manager	Created by the Session Controller when the first MAG service is created in a context.	Packet processing card		Same as VPN Mgr.	
	Sends and receives PMIP control messages (PBU/PBA).				
	Adds an NPU flow to receive MIPv6 PBA packets. This flow is identical to the flow used in the HAMgr.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	<p>Maintains the Binding Update List used to keep track of the mobile node's bindings:</p> <ul style="list-style-type: none"> • MN-ID • APN • Home Network Prefix and prefix length • IPv6 LMA address • IPv4 Home Address 				
	Originates PBU-based on trigger received from the Session Manager during error conditions.				
	Receives PBA and forwards it to Session Manager.				
	Supports debugging facility – “magmgr” and “mobile-ipv6”.				
Mobility Management Entity Demux Manager (MMEDemux)	Started as part of MME service creation procedure. There will be only one MME DEMUX MGR in the chassis.	Packet processing card	Packet processing card (see NOTE)	0	Any (see NOTE)
	Distributes incoming S1-MME SCTP connections to MME Mgrs in the system.				
	Remains aware of all the active MME services in the system.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the First active packet processing card but should not be created on the same packet processing card that has MME Manager.				
Mobility Management Entity Manager (MMEMgr)	Starts when an MME service configuration is detected. There can be multiple instances of this task for load sharing. All MME Managers will have all the Active MME Services configured and will be identical in configuration and capabilities.	Active packet processing card	Active Active packet processing card (see NOTE)	0	Any (see NOTE)
	Runs the SCTP protocol stack.				
	Handles the SCTP associations.				
	Maintains TA List.				
	Manage eNodeB databases.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	Provides nodal functions for S1-MME protocol.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card. The MMEMGRs should not be started on a packet processing card which has the MME DEMUX MGR started.				
PCC Bindmux Manager (PCC Demux)	Started as part of PCC service creation procedure. There will be only one BindMux MGR in the chassis.	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)
	Handles multiplexing of the sessions across the available PCCMgrs along with the session binding functions				
	Monitors load on PCCMgrs.				
	Distributes incoming IP-CAN connections across PCC Mgrs in the system.				
	Performs session binding; binds IP-CAN/Gateway session with the AF-Session.				
	Ensures all messaging for an IMSI across various interfaces is directed towards the selected PCCMgr.				
	Remains aware of all the active PCC services in the system.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card but should not be created on the same packet processing card that has PCC Manager.				
SGSN GPRS Tunneling Protocol Control message Manager (SGTPCMgr)	Created by the Session Controller for each VPN context in which an SGSN service is configured.	Active packet processing card	Active Demux packet processing card (see NOTE)	0	Any (see NOTE)
	Terminates Gn/Gp and GTP-U interfaces from peer GGSNs and SGSNs for SGSN Services.				
	Terminates GTP-U interfaces from RNCs for IuPS Services.				
	Controls standard ports for GTP-C and GTP-U.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	Processes and distributes GTP-traffic received from peers on these ports.				
	Performs all node level procedures associated with Gn/Gp interface.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.				
SGSN Master Manager (MMgr)	Created upon provisioning of SS7RDs/SCCP-NWs/etc. The Session Controller provides the initial system configuration which includes a detailed description of each distributed protocol layer, its resources sets, and a list of its service user protocol layers and service provider protocol layers.	Active packet processing card	Active Demux packet processing card (see NOTE)	0	Any (see NOTE)
	Runs as a single instance.				
	Handles nodal SS7, Iu, and Gb functionality.				
	Implements master LinkMgr functionality for SS7 route status aggregation.				
	Implements master LinkMgr functionality for RNC and BSC status aggregation.				
	NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.				
SS7 Link Manager (LinkMgr)	Created by the Session Controller when the first SS7RD (routing domain) is activated, the LinkMgr performs the following functions:	Any Active packet processing card not running an MMgr	Any Active non-demux packet processing card		
	Multi-instanced for redundancy and scaling purposes.				
	Provides SS7 and Gb connectivity to the platform.				

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
	Routes per subscriber signalling across the SS7 (including Iu) and Gb interfaces to the SessMgr.				
Standard Routing Database (SRDB)	Eight SRDBs are created by the Session Controller when Content Filtering in the Enhanced Charging Service is enabled. A minimum of two packet processing cards are required to initiate these eight tasks.	Peer SRDBs evenly distributed across packet processing cards			
	Receives the static database from the session controller. Each SRDB task loads two database volumes (one primary and one secondary). The SRDB task also stores the static DB.				
	Rrates and categorizes the URL based on the DB volumes and CSI (category set Index) stored on it.				
	Performs peer loading in case its peer fails. If both the SRDB task and its peer fail, the session controller performs the loading.				

Table 48. ASR 5x00 Platform Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Card-Slot-Port Controller (CSPCTRL)	Manages physical chassis components.	MIO/UMIO	MIO/UMIO	0	0
Messenger Daemon (MSGD)	Implements the Name Service and related functions for the internal message passing system.	All	All	All	All
Name Service Controller (NSCONTROL)	As part of the Messenger process, provides a reliable channel for tasks to send control messages to the Messenger Daemon.	All	All	All	All
Daughter Card Controller (DCARDCTRL)	Spawns daughter card managers during system initialization and monitors daughter card managers during system steady state execution. It also spawns daughter card managers whenever a daughter card manager task fails.	Active MIO/UMIO	Active MIO/UMIO	0	0
Daughter Card Manager	Responsible for managing IPsec Security Associations for AH- and ESP-based sessions.	All packet processing	All packet processing	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
(DCARDMGR)	Interfaces with the on-board hardware accelerated cryptographic chip which executes cryptographic algorithms associated with the given IPSec Security Associations.	cards	cards		
Distributed Host Manager (DHMGR)	Started automatically on each CPU by SITPARENT.	All	All	All	All
	Coordinates establishment of locally terminated TCP, SCTP, and UDP connections on behalf of multi-instanced tasks such as Diameter endpoints among SESSMGR tasks.				
Driver Controller (DRVCTRL)	Centralizes access to many of the system device drivers. It also performs temperature and voltage monitoring.	MIO/UMIOs	MIO/UMIOs	0	0
Hard Drive Controller (HDCTRL)	Controls and manages the drive array spanning the SMCsMIO/UMIOs.	MIO/UMIOs	MIO/UMIOs	0	0
IPSec Controller (IPSECCTRL)	Started by SIT on system startup regardless of configuration.	Active MIO/UMIO	Active MIO/UMIO	0	0
	Starts IPSECMGR tasks based on configuration and maintains its list for task recovery.				
	Receives and maintains user configuration for IPSec.				
	Manages the configured IPSec crypto maps and its assignment to IPSECMGRs.				
	Interfaces with the VPNMGR task for required IPSec configuration parameters such as IP Access-lists, IP pools, interface addresses, and interface state notifications.				
IPSec Manager (IPSECMGR)	Created by the Session Controller, establishes and manages secure IKEv1, IKEv2 and IPSec data tunnels.	Packet processing card	Packet processing card	All	All
Recovery Control Task (RCT)	Monitors tasks/managers/facilities across the system and performs recovery in the event of a failure.	MIO/UMIOs	MIO/UMIOs	0	0
Shared Configuration Task (SCT)	Performs the redundant storage of configuration information and other state information in an in-memory database.	MIO/UMIOs	MIO/UMIOs	0	0
Switch Fabric Task (SFT)	Monitors the switch fabric and the gigabit Ethernet control plane.	Packet processing cards	Packet processing cards	0	0
Utilities Configuration Manager (UCM)	DHCPD, DNS, FTPD, INETD, NTPD, PING, RLOGIN, SFTPD, SFTP-SERVER, SNMPD, SSH, SSHD, TELNET, TELNETD, TFTP, TRACEROUTE	Active MIO/UMIO	Active MIO/UMIO	0	0

Table 49. ASR 5x00 Management Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Bulk Statistic Manager (BULKSTAT)	Performs a periodic statistic polling/gathering function (bulk statistics) and handles the transfer of this data to external management systems.	Active MIO/UMIO	Active MIO/UMIO	0	0
Event Log Daemon (EVLOGD)	Handles event logging functions including the interface to external syslogd servers and the internal event logs.	Active MIO/UMIO	Active MIO/UMIO	0	0
ORB Service (ORBS)	The ORBS task is also known as the ORB Element Manager (ORBEM). Application Servers (EMS) request ORBS to perform Element Management Functions on the system using secure IIOP. ORBS then interacts with concerned Controller Tasks to execute the function. The response/errors from the execution is interpreted, formulated into EMF response, and handed over to Application Server (EMS).	Active MIO/UMIO	Active MIO/UMIO	0	0
ORB Notification Service (ORBNS)	Notifies the Application Servers (EMS) of event occurrences.	Active MIO/UMIO	Active MIO/UMIO	0	0
	Registers such Application Servers (EMS) and subscribes them to associated event types.				
	As the events occur, the concerned Controller Task notifies ORBS (ORBEM), which then notifies the subscribing Application Servers (EMS).				
Session Trace Collection Task (SESSTRC)	Implements the standards-based session trace functionality.	Active MIO/UMIO	Active MIO/UMIO	0	0
	Manages both CLI and signaling-based subscriber traces. It collects messages to be traced and generates trace files as needed. It uploads trace files to the Trace Collection Entity as needed.				
Simple Network Management Protocol (SNMP)	Handles inboard SNMP operations if configured, and sends SNMP notifications (traps) if enabled.	Active MIO/UMIO	Active MIO/UMIO	0	0
Threshold Server (THRESHOLD)	Handles monitoring of threshold crossing alerts, if configured. Polls the needed statistics/variables, maintains state, and generates log messages/SNMP notification of threshold crossings.	Active MIO/UMIO	Active MIO/UMIO	0	0

Table 50. ASR 5500 Fabric Management Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Fabric Controller (AFCTRL)	Responsible for the overall management of the system fabric. Manages the pool of Rendezvous Destinations and coordinates fabric recovery by the AFMGR proclcts after a fault. There is a single AFCTRL instance per system running on the active MIO/UMIO.	Active MIO/UMIO	Active MIO/UMIO	0	0
Fabric Manager (AFMGR)	Responsible for the management of fabric resources on a particular card. There is one AFMGR on every CPU that is responsible for one or more fabric access processors (FAPs) or fabric elements (FEs). AFMGR support recovery but not migration.	All DPC/UDPCs, MIO/UMIOs, FSCs	All DPC/UDPCs, MIO/UMIOs, FSCs	0	0
AFIO Driver	Responsible for the direct configuration of the fabric chipset. AFIO supports non-messenger interprocess communication (IPC) with the local AFMGR and with other local AFIO instances	All DPC/UDPCs, MIO/UMIOs, FSCs	All DPC/UDPCs, MIO/UMIOs, FSCs	0	0
Simple Network Management Protocol (SNMP)	Handles inboard SNMP operations if configured, and sends SNMP notifications (traps) if enabled.	Active MIO/UMIO	Active MIO/UMIO	0	0

Appendix C

Access Control Lists

This segment describes system support for access control lists and explains how they are configured. The product administration guides provide product administration guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model before using the procedures described below.

 **Important:** You do not require a license to configure ACLs. However, the number of ACLs configured may impact performance significantly.

This segment contains the following sections:

- [Understanding ACLs](#)
- [Configuring ACLs on the System](#)
- [Applying IP ACLs](#)

 **Important:** Not all commands and keywords/variables may be available. Availability depends on the platform type.

Overview

IP access lists, commonly known as access control lists (ACLs), control the flow of packets into and out of the system. They are configured on a per-context basis and consist of “rules” (ACL rules) or filters that control the action taken on packets that match the filter criteria. Once configured, an ACL can be applied to any of the following:

- An individual interface
- All traffic facilitated by a context (known as a policy ACL)
- An individual subscriber
- All subscriber sessions facilitated by a specific context

Separate ACLs may be created for IPv4 and IPv6 access routes.

Understanding ACLs

This section discusses the two main aspects to ACLs on the system:

- [Rule\(s\)](#)
- [Rule Order](#)

 **Important:** Refer to *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* in the *Command Line Interface Reference* for the full command syntax.

Rule(s)

A single ACL consists of one or more ACL rules. Each rule is a filter configured to take a specific action when packets matching specific criteria. Up to 128 rules can be configured per ACL.

 **Important:** Configured ACLs consisting of no rules imply a “deny any” rule. The **deny** action and **any** criteria are discussed later in this section. This is the default behavior for an empty ACL.

Each rule specifies the action to take when a packet matches the specifies criteria. This section discusses the rule actions and criteria supported by the system.

Actions

ACLs specify that one of the following actions can be taken on a packet that matches the specified criteria:

- **Permit:** The packet is accepted and processed.
- **Deny:** The packet is rejected.
- **Redirect:** The packet is forwarded to the specified next-hop address through a specific system interface or to the specified context for processing.

 **Important:** Redirect rules are ignored for ACLs applied to specific subscribers or all subscribers facilitated by a specific context, or APN for UMTS subscribers.

Criteria

Each ACL consists of one or more rules specifying the criteria that packets will be compared against.

The following criteria are supported:

- **Any:** Filters all packets
- **Host:** Filters packets based on the source host IP address
- **ICMP:** Filters Internet Control Message Protocol (ICMP) packets
- **IP:** Filters Internet Protocol (IP) packets
- **Source IP Address:** Filter packets based on one or more source IP addresses
- **TCP:** Filters Transport Control Protocol (TCP) packets
- **UDP:** Filters User Datagram Protocol (UDP) packets

Each of the above criteria are described in detail in the sections that follow.

 **Important:** The following sections contain basic ACL rule syntax information. Refer to *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* in the *Command Line Interface Reference* for the full command syntax.

- **Any:** The rule applies to all packets.
- **Host:** The rule applies to a specific host as determined by its IP address.
- **ICMP:** The rule applies to specific Internet Control Message Protocol (ICMP) packets, Types, or Codes. ICMP type and code definitions can be found at www.iana.org (RFC 3232).
- **IP:** The rule applies to specific Internet Protocol (IP) packets or fragments.
- **IP Packet Size Identification Algorithm:** The rule applies to specific Internet Protocol (IP) packets identification for fragmentation during forwarding.

This configuration is related to the “IP Identification field” assignment algorithm used by the system, when subscriber packets are being encapsulated (such as Mobile IP and other tunneling encapsulation). Within the system, subscriber packet encapsulation is done in a distributed way and a 16-bit IP identification space is divided and distributed to each entity which does the encapsulation, so that unique IP identification value can be assigned for IP headers during encapsulation.

Since this distributed IP Identification space is small, a non-zero unique identification will be assigned only for those packets which may potentially be fragmented during forwarding (since the IP identification field is only used for reassembly of the fragmented packet). The total size of the IP packet is used to determine the possibility of that packet getting fragmented.

- **Source IP Address:** The rule applies to specific packets originating from a specific source address or a group of source addresses.
- **TCP:** The rule applies to any Transport Control Protocol (TCP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. TCP port numbers definitions can be found at www.iana.org

- **UDP:** The rule applies to any User Datagram Protocol (UDP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. UDP port numbers definitions can be found at www.iana.org.

Rule Order

A single ACL can consist of multiple rules. Each packet is compared against each of the ACL rules, in the order in which they were entered, until a match is found. Once a match is identified, all subsequent rules are ignored.

Additional rules can be added to an existing ACL and properly ordered using either of the following options:

- Before
- After

Using these placement options requires the specification of an existing rule in the ACL and the configuration of the new rule as demonstrated by the following flow:

```
[ before | after ] { <existing_rule> }
```

Configuring ACLs on the System

This section describes how to configure ACLs.

 **Important:** This section provides the minimum instruction set for configuring access control list on the system. For more information on commands that configure additional parameters and options, refer to *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide an access control list facility to subscribers:

- Step 1** Create the access control list by following the example configuration in the [Creating ACLs](#) section.
- Step 2** Specify the rules and criteria for action in ACL list by following the example configuration in the [Configuring Action and Criteria for Subscriber Traffic](#) section.
- Step 3** *Optional.* The system provides an “undefined” ACL that acts as a default filter for all packets into the context. The default action is to “permit all”. Modify the default configuration for “unidentified” ACLs for by following the example configuration in the [Configuring an Undefined ACL](#) section.
- Step 4** Verify your ACL configuration by following the steps in the [Verifying the ACL Configuration](#) section.
- Step 5** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Creating ACLs

To create an ACL, enter the following command sequence from the Exec mode of the system CLI:

```
configure
context <acl_ctxt_name> [ -noconfirm ]
    ( ip | ipv6 ) access-list <acl_list_name>
end
```

Notes:

- The maximum number of ACLs that can be configured per context is limited by the amount of available memory in the VPN Manager software task. Typically, the maximum is less than 200.

Configuring Action and Criteria for Subscriber Traffic

To create rules to deny/permit the subscriber traffic and apply the rules after or before action, enter the following command sequence from the Exec mode of the system CLI:

```
configure
context <acl_ctxt_name> [ -noconfirm ]
    ( ip | ipv6 ) access-list <acl_list_name>
        deny { <ip_address> | any | host | icmp | ip | log | tcp | udp }
        permit { <ip_address> | any | host | icmp | ip | log | tcp | udp }
        after { deny | permit | readdress | redirect }
        before { deny | permit | readdress | redirect }
    end
```

Notes:



Caution: The system does not apply a “deny any” rule, unless it is specified in the ACL. This behavior can be changed by adding a “deny any” rule at the end of the ACL.

- The maximum number of rules that can be configured per ACL varies depending on how the ACL is to be used. For more information, refer to *Engineering Rules*.
- Use the information provided in the [Actions](#) and [Criteria](#) to configure the rules that comprise the ACL. For more information, refer to *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* in the *Command Line Interface Reference*.

Configuring an Undefined ACL

As discussed previously the system uses an “undefined” ACL mechanism for filtering the packet(s) in the event that an ACL that has been applied is not present. This scenario is likely the result of a mis-configuration such as the ACL name being mis-typed during the configuration process.

For these scenarios, the system provides an “undefined” ACL that acts as a default filter for all packets into the context. The default action is to “permit all”.

To modify the default behavior for unidentified ACLs, use the following configuration:

```
configure
context <acl_ctxt_name> [-noconfirm]
    access-list undefined { deny-all | permit-all }
end
```

Notes:

- Context name is the name of the context containing the “undefined” ACL to be modified. For more information, refer to *Context Configuration Mode Commands* in the *Command Line Interface Reference*.

Verifying the ACL Configuration

To verify the ACL configuration, enter the Exec mode **show { ip | ipv6 } access-list** command.

The following is a sample output of this command. In this example, an ACL named `acl_1` was configured.

```
ip access list acl_1
  deny host 10.2.3.4
  deny ip any host 10.2.3.4
  permit any 10.2.4.4
1 ip access-lists are configured.
```

Applying IP ACLs

Once an ACL is configured, it must be applied to take effect.

 **Important:** All ACLs should be configured and verified according to the instructions in the [Configuring ACLs on the System](#) prior to beginning these procedures. The procedures described below also assume that the subscribers have been previously configured.

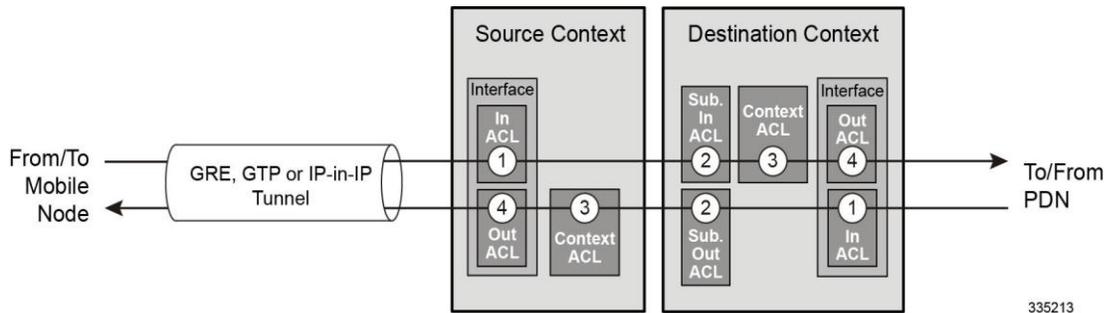
As discussed earlier, you can apply an ACL to any of the following:

- [Applying an ACL to an Individual Interface](#)
- [Applying an ACL to All Traffic Within a Context](#) (known as a policy ACL)
- [Applying an ACL to an Individual Subscriber](#)
- [Applying a Single ACL to Multiple Subscribers](#)
- [Applying a Single ACL to Multiple Subscribers via APNs](#) (for 3GPP subscribers only)

 **Important:** ACLs must be configured in the same context in which the subscribers and/or interfaces to which they are to be applied. Similarly, ACLs to be applied to a context must be configured in that context.

If ACLs are applied at multiple levels within a single context (such as an ACL is applied to an interface within the context and another ACL is applied to the entire context), they will be processed as shown in the following figure and table.

Figure 14. ACL Processing Order



335213

Table 51. ACL Processing Order Descriptions

Packet coming from the mobile node to the packet data network (left to right)	
Order	Description
1	An inbound ACL configured for the receiving interface in the Source Context is applied to the tunneled data (such as the outer IP header). The packet is then forwarded to the Destination Context.
2	An inbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied.
3	A context ACL (policy ACL) configured in the Destination Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Destination Context through which the packet is being forwarded, is applied.
Packet coming from the packet data network to the mobile node (right to left)	
Order	Description
1	An inbound ACL configured for the receiving interface configured in the Destination Context is applied.
2	An outbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied. The packet is then forwarded to the Source Context.
3	A context ACL (policy ACL) configured in the Source Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Source Context through which the packet is being forwarded, is applied to the tunneled data (such as the outer IP header).

In the event that an IP ACL is applied that has not been configured (for example, the name of the applied ACL was configured incorrectly), the system uses an “undefined” ACL mechanism for filtering the packet(s).

This section provides information and instructions for applying ACLs and for configuring an “undefined” ACL.

Applying an ACL to an Individual Interface

This section provides information and instructions for applying one or more ACLs to an individual interface configured on the system.

 **Important:** This section provides the minimum instruction set for applying the ACL list to an interface on the system. For more information on commands that configure additional parameters and options, refer to *Ethernet Interface Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide ACL facility to subscribers:

- Step 1** Apply the configured access control list by following the example configuration in the [Applying the ACL to an Interface](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration on an Interface](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying the ACL to an Interface

To apply the ACL to an interface, use the following configuration:

```
configure
  context <acl_ctxt_name> [ -noconfirm ]
    interface <interface_name>
      { ip | ipv6 } access-group <acl_list_name> { in | out } [ <preference> ]
    end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration on an Interface

This section describes how to verify the ACL configuration.

Step 1 In the Exec Mode, enter the following command:

```
show configuration context context_name
```

context_name is the name of the context containing the interface to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
context context_name
  ip access-list acl_name
    deny host ip_address
    deny ip any host ip_address
  exit
  ip access-group access_group_name
  service-redundancy-protocol
  exit
  interface interface_name
    ip address ip_address/mask
  exit
  subscriber default
  exit
  aaa group default
  exit
  gtpv group default
  end
```

Applying an ACL to All Traffic Within a Context

This section provides information and instructions for applying one or more ACLs to a context configured within a specific context on the system. The applied ACLs, known as policy ACLs, contain rules that apply to all traffic facilitated by the context.

 **Important:** This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Context Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured ACL as described in the [Applying the ACL to a Context](#) section.

Step 2 Verify that ACL is applied properly on interface as described in the [Verifying the ACL Configuration in a Context](#) section.

- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying the ACL to a Context

To apply the ACLs to a context, use the following configuration:

```
configure
  context <acl_ctxt_name> [-noconfirm]
    { ip | ipv6 } access-group <acl_list_name> [ in | out ] [ <preference> ]
  end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The context-level ACL are applied only to outgoing packets. The **in** and **out** keywords are deprecated and are only present for backward compatibility.
- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration in a Context

To verify the ACL configuration:

- Step 1** Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
    exit
    ip access-group access_group_name
    service-redundancy-protocol
    exit
  interface interface_name
    ip address ip_address/mask
  exit
  subscriber default
  exit
```

```

aaa group default
  exit
gtp group default
  end

```

Applying an ACL to a RADIUS-based Subscriber

IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To apply an ACL to a RADIUS-based subscriber, use the **Filter-Id** attribute.

For more details on this attribute, if you are using StarOS 12.3 or an earlier release, refer to the *AAA and GTPP Interface Administration and Reference*. If you are using StarOS 14.0 or a later release, refer to the *AAA Interface Administration and Reference*.

This section provides information and instructions for applying an ACL to an individual subscriber whose profile is configured locally on the system.



Important: This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Subscriber Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by following the example configuration in the [Applying an ACL to an Individual Subscriber](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to an Individual Subscriber](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying an ACL to an Individual Subscriber

To apply the ACL to an individual subscriber, use the following configuration:

```

configure
  context <acl_ctxt_name> [ -noconfirm ]
    subscriber name <subs_name>
      { ip | ipv6 } access-group <acl_list_name> [ in | out ]
    end

```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.

- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to an Individual Subscriber

These instructions are used to verify the ACL configuration.

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the subscriber *subs1* to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
context context_name
  ip access-list acl_name
    deny host ip_address
    deny ip any host ip_address
  exit
  ip access-group access_group_name
  service-redundancy-protocol
  exit
  interface interface
    ip address ip_address/mask
  exit
  subscriber default
  exit
  subscriber name subscriber_name
    ip access-group access_group_name in
    ip access-group access_group_name out
  exit
  aaa group default
  exit
  gtp group default
  exit
  content-filtering server-group cfsq_name
    response-timeout response_timeout
    connection retry-timeout retry_timeout
  end
```

Applying a Single ACL to Multiple Subscribers

As mentioned in the previous section, IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. The following table describes these functions.

Table 52. Functions Used to Provide “Default” Subscriber Attributes

Function	Description
Subscriber Named <i>default</i>	<p>Within each context, the system creates a subscriber called <i>default</i>. The profile for the subscriber named <i>default</i> provides a configuration template of attribute values for subscribers authenticated in that context. Any subscriber attributes that are not included in a RADIUS-based subscriber profile is configured according to the values for those attributes as defined for the subscriber named <i>default</i>.</p> <p>NOTE: The profile for the subscriber named <i>default</i> is <u>not</u> used to provide missing information for subscribers configured locally.</p>
default subscriber Command	<p>This command in the PDSN, FA, and HA service Configuration modes specifies a profile from a subscriber named something other than <i>default</i> to use a configuration template of attribute values for subscribers authenticated in that context.</p> <p>This command allows multiple services to draw “default” subscriber information from multiple profiles.</p>

When configured properly, the functions described in the table above could be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the “default” profile.

Applying an ACL to the Subscriber Named default

This section provides information and instructions for applying an ACL to the subscriber named *default*.

 **Important:** This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Subscriber Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by following the example configuration in the [Applying an ACL to the Subscriber Named default](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Applying an ACL to the Subscriber Named default](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying an ACL to the Subscriber Named default

To apply the ACL to the subscriber named *default*, use the following configuration:

```
configure
  context <acl_ctxt_name> [ -noconfirm ]
    subscriber name <subs_name>
      { ip | ipv6 } access-group <acl_list_name> [ in | out ]
    end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to the Subscriber Named default

These instructions are used to verify the ACL configuration.

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the subscriber default to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
    exit
    ip access-group access_group_name
    service-redundancy-protocol
    exit
    interface interface
      ip address ip_address/mask
    exit
    subscriber name default
      ip access-group access_group_name in
      ip access-group access_group_name out
    exit
  aaa group default
    exit
  gtpv6 group default
```

```

exit
content-filtering server-group cfsg_name
response-timeout response_timeout
connection retry-timeout retry_timeout
end

```

Applying an ACL to Service-specified Default Subscriber

This section provides information and instructions for applying an ACL to the subscriber to be used as the “default” profile by various system services.

 **Important:** This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Subscriber Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by following the example configuration in the [Applying an ACL to Service-specified Default Subscriber](#) section.
- Step 2** Verify that the ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to Service-specified Default Subscriber](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying an ACL to Service-specified Default Subscriber

To apply the ACL to a service-specified Default subscriber, use the following configuration:

```

configure
context <acl_ctxt_name> [ -noconfirm ]
    { pdsn-service | fa-service | ha-service } <service_name>
        default subscriber <svc_default_subs_name>
    exit
subscriber name <svc_default_subs_name>
    { ip | ipv6 } access-group <acl_list_name> [ in | out ]
end

```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.

- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to Service-specified Default Subscriber

To verify the ACL configuration.

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the service with the default subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
context context_name
  ip access-list acl_name
    deny host ip_address
    deny ip any host ip_address
  exit
  ip access-group access_group_name
  interface interface
    ip address ip_address/mask
  exit
  subscriber default
  exit
  subscriber name subscriber_name
    ip access-group access_group_name in
    ip access-group access_group_name out
  exit
  pdsn-service service_name
    default subscriber subscriber_name
  end
```

Applying an ACL to Multiple Subscriber via APNs

If IP ACLs are applied to subscribers via attributes in their profile, the subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates for GGSN subscribers. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

This section provides information and instructions for applying an ACL to an APN template.

 **Important:** This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Subscriber Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by following the example configuration in the [Applying an ACL to Multiple Subscriber via APNs](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to APNs](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to *Verifying and Saving Your Configuration*.

Applying an ACL to Multiple Subscriber via APNs

To apply the ACL to multiple subscribers via APN, use the following configuration:

```
configure
  context <dest_context_name> [-noconfirm]
    apn <apn_name>
      { ip | ipv6 } access-group <acl_list_name> [ in | out ]
    end
```

Notes:

- The ACL to be applied must be in the destination context of the APN (which can be different from the context where the APN is configured).
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to APNs

To verify the ACL configuration:

- Step 1** Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the APN *apn1* having *default* subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
  context context_name
    ip access-list acl_name
      deny host ip_address
      deny ip any host ip_address
    exit
    ip access-group access_group_name
  interface interface
```

```
        ip address ip_address/mask
    exit
subscriber default
    exit
apn apn_name
    ip access-group access_group_name in
    ip access-group access_group_name out
end
```


Appendix D

Congestion Control

This segment describes the Congestion Control feature. It covers the following topics:

- [Overview](#)
- [Configuring Congestion Control](#)

Overview

Congestion Control monitors the system for conditions that could potentially degrade performance when the system is under heavy load. Typically, these conditions are temporary (for example, high CPU or memory utilization) and are quickly resolved. However, continuous or large numbers of these conditions within a specific time interval may impact the system's ability to service subscriber sessions. Congestion control helps identify such conditions and invokes policies for addressing the situation.

Congestion control operation is based on configuring the following:

- **Congestion Condition Thresholds:** Thresholds dictate the conditions for which congestion control is enabled and establishes limits for defining the state of the system (congested or clear). These thresholds function in a way similar to operation thresholds that are configured for the system as described in the *Thresholding Configuration Guide*. The primary difference is that when congestion thresholds are reached, a service congestion policy and an SNMP trap (starCongestion) are generated.

A threshold tolerance dictates the percentage under the configured threshold that must be reached in order for the condition to be cleared. An SNMP trap, starCongestionClear, is then triggered.

- **Port Utilization Thresholds:** If you set a port utilization threshold, when the average utilization of all ports in the system reaches the specified threshold, congestion control is enabled.
- **Port-specific Thresholds:** If you set port-specific thresholds, when any individual port-specific threshold is reached, congestion control is enabled system-wide.
- **Service Congestion Policies:** Congestion policies are configurable for each service. These policies dictate how services respond when the system detects that a congestion condition threshold has been crossed.



Important: This section provides the minimum instruction set for configuring congestion control. Commands that configure additional interface or port properties are provided in *Subscriber Configuration Mode* in the *Command Line Interface Reference*. Always refer to the Administration Guides for all of the licensed products running on this platform for additional configuration information with respect to congestion control. Congestion control functionality varies based on product and StarOS version.

For the MME three levels of congestion control thresholds are supported – critical, major and minor. By default only the critical threshold is supported for other products. SNMP traps also support major and minor congestion control thresholds. A set of **congestion-action-profile** commands allows an operator to establish additional actions to be taken for specific thresholds and threshold levels.

Configuring Congestion Control

To configure Congestion Control functionality:

- Step 1** Configure congestion control thresholds as described in the [Configuring the Congestion Control Threshold](#) section.
- Step 2** Configure service congestion policies as described in the [Configuring Service Congestion Policies](#) section.
- Step 3** Enable redirect overload policies as described in the [Enabling Congestion Control Redirect Overload Policy](#) section.
- Step 4** Configure disconnecting subscribers based on call or inactivity time as described in the [Disconnecting Subscribers Based on Call or Inactivity Time](#) section.
- Step 5** Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring the Congestion Control Threshold

To configure congestion control threshold, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure
    congestion-control threshold max-sessions-per-service-utilization <percent>
    congestion-control threshold tolerance <percent>
end
```

Notes:

- There are numerous threshold parameters. See *Global Configuration Mode* in the *Command Line Interface Reference* for more information.
- The tolerance is the percentage under a configured threshold that dictates the point at which the condition is cleared.
- Multiple levels of congestion thresholds – critical, major and minor – are supported for various types of congestion control thresholds. If a threshold level is not specified, the default is critical. Currently, major and minor thresholds are only supported for the MME. The **congestion-action-profile** command under **lte-policy** defines the action to be taken when thresholds are exceeded. See *Global Configuration Mode Commands, LTE Policy Configuration Mode Commands* and *Congestion Action Profile Configuration Mode Commands* in the *Command Line Interface Reference* for more information.
- Repeat this configuration as needed for additional thresholds.

Configuring Service Congestion Policies

To create a congestion control policy, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure
    congestion-control policy <service> action { drop | none | redirect | reject }
end
```

Notes:

- When the redirect action occurs for PDSN services, the PDSN responds to the PCF with a reply code of 136, “unknown PDSN address” along with the IP address of an alternate PDSN.
- **redirect** is not available for PDIF. The default action for PDIF is “none.”
- When the redirect action occurs for HA services, the system responds to the FA with a reply code of 136, “unknown home agent address”.
- **redirect** cannot be used in conjunction with GGSN services.
- **redirect** is not available for the Local Mobility Anchor (LMA) service.
- When setting the action to **reject**, the reply code is 130, “insufficient resources”.
- For the GGSN, the reply code is 199, “no resources available”.
- For the MME, **redirect** is not available.
- For the MME, create action profiles for optional major and minor thresholds using the **congestion-action-profile** command under **lte-policy** in the Global Configuration mode.
- For the MME, you can specify <service> as **critical**, **major** or **minor** to set a policy and associate an action-profile for the respective threshold. See *Global Configuration Mode Commands* in the *Command Line Interface Reference* for more information.

Configuring Overload Reporting on the MME (ASR 5000 only)

When an overload condition is detected on an MME and the report-overload keyword is enabled in the **congestion-control policy** command, the system reports the condition to a specified percentage of eNodeBs and proceeds to take the configured action on incoming sessions. To create a congestion control policy with overload reporting, apply the following example configuration:

```
configure
    congestion-control policy mme-service action report-overload reject-new-
sessions enodeb-percentage <percentage>
end
```

Notes:

- Other overload actions include **permit-emergency-sessions** and **reject-non-emergency-sessions**.

Enabling Congestion Control Redirect Overload Policy

To create a congestion control policy and configure a redirect overload policy for the service, apply the following example configuration:



Important: Redirect is not available on PDIF or MME for this release.

```
configure
  congestion-control
  context <context_name>
    {service_configuration_mode}
    policy overload redirect address
  end
```

Notes:

- *Optional:* If the congestion control policy action was configured to **redirect**, then a redirect overload policy must be configured for the service(s) that are affected.
- There are several service configuration modes that you can configure. See the *Command Line Interface Reference* for a complete list of modes.
- You can set various options for redirection. See the *Command Line Interface Reference* for more information.
- Repeat this configuration example to configure overload policies for additional services configured in the same context.

Verify the Service Overload Policies

To verify that the service overload policies were properly configured enter the following command in the Exec Mode:

```
show service_type name service_name
```

This command lists the entire service configuration. Verify that the information displayed for the “Overload Policy” is accurate.

Repeat this configuration example to configure additional services in other contexts.

Verify the Congestion Control Configuration

To verify Congestion Control Configuration enter the **show congestion-control configuration** command in the Exec Mode.

The following output is a concise listing of all threshold and policy configurations showing multi-level Critical, Major and Minor threshold parameters:

```
Congestion-control: enabled

Congestion-control Critical threshold parameters
  system cpu utilization:          80%
  service control cpu utilization: 80%
  system memory utilization:      80%
  message queue utilization:      80%
  message queue wait time:       10 seconds
  port rx utilization:            80%
  port tx utilization:            80%
  license utilization:            100%
  max-session-per-service utilization: 100%
  tolerance limit:                10%

Congestion-control Critical threshold parameters
  system cpu utilization:          80%
  service control cpu utilization: 80%
  system memory utilization:      80%
  message queue utilization:      80%
  message queue wait time:       10 seconds
  port rx utilization:            80%
  port tx utilization:            80%
  license utilization:            100%
  max-session-per-service utilization: 100%
  tolerance limit:                10%
  session disconnect percent:     5%
  iterations-per-stage:           8

Congestion-control Policy
  pdsn-service: none
  hsgw-service: none
  ha-service: none
  ggsn-service: none
  closedrp-service: none
  lns-service: none
  cscf-service: reject
  pdif-service: none
  fng-service: none
  sgsn-service: none
  mme-service: drop
  asngw-service: none
  asnpc-service: none
  phsgw-service: none
  phspc-service: none
  mipv6ha-service: none
  lma-service: none
  saegw-service: none
  sgw-service: none
  pgw-service: none
  hnbgw-service: none
  pcc-policy-service: none
  pcc-quota-service: none
  pcc-af-service: none
  ipsg-service: none
```

The primary threshold to observe is *license utilization*. This threshold is defaulted to 80%. Overload controls on the system enables the Congestion-control Policy when the system has only 80% of the licenses used. The overload condition will not clear until the utilization drops below the tolerance limit setting. The tolerance limit is defaulted to 10%. If the system goes into overload due to license utilization (threshold at 80%), the overload condition will not clear until the license utilization reaches 70%.

The system may go into overload if threshold settings are set too low and congestion control is enabled. You will need to review all threshold values and become familiar with the settings.

Since the recommendation for license utilization overload threshold is 100%, you should enable a license threshold alarm at 80%. An alarm is then triggered when the license utilization hits 80%. When the congestion-control policy setting is set to **drop**, the system drops incoming packets containing new session requests.

 **Important:** For additional information on configuring the alarm threshold, refer to the *Threshold Configuration Guide*.

Verify MME Congestion Action Profiles

To verify MME multilevel congestion action profiles, run the following Exec mode command:

```
show lte-policy congestion-action-profile { name <profile_name> | summary }
```

Disconnecting Subscribers Based on Call or Inactivity Time

During periods of heavy system load, it may be necessary to disconnect subscribers in order to maintain an acceptable level of system performance. You can establish thresholds to select subscribers to disconnect based on the length of time that a call has been connected or inactive.

To enable overload disconnect for the currently selected subscriber, use the following configuration example:

```
configure
  context <context_name>
    subscriber name <subscriber_name>
      default overload-disconnect threshold inactivity-time <dur_thresh>
      default overload-disconnect threshold connect-time <dur_thresh>
    end
```

To disable the overload disconnect feature for this subscriber, use the following configuration example:

```
configure
  context <context_name>
    subscriber <subscriber_name>
      no overload-disconnect {[threshold inactivity-time] | [threshold connect-time]}
    end
```

Notes:

- **overload-disconnect** is not supported for the Call Session Control Function (CSCF) service.

Appendix E

Content Service Steering

This segment provides information on configuring Content Service Steering (CSS). The product administration guides provide provides examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product administration guide, before using the procedures described below.

 **Important:** Internal CSS is a generic feature, if an ECSv2 license is installed on your system, internal CSS can be enabled. A separate license is not required to enable internal CSS. Contact your local Cisco account representative for information on how to obtain a license.

This segment contains the following topics:

- [Overview](#)
- [Configuring Internal Content Service Steering](#)

Overview

Content Service Steering (CSS) selectively directs subscriber traffic to In-line services internal to the system based on data content presented by mobile subscribers. CSS is a broad term that includes features such as NAT, HTTP redirection, and DNS redirection.

CSS uses Access Control Lists (ACLs) to redirect subscriber traffic flows. ACLs control the flow of packets into and out of the system. ACLs consist of “rules” (ACL rules) or filters that control the action taken on packets matching the filter criteria.

ACLs are configurable on a per-context basis and applies to a subscriber through either a subscriber profile (or an APN profile in the destination context). For additional information, refer to *Access Control Lists*.

Configuring Internal Content Service Steering

To configure and activate a single CSS service for redirecting all of a subscriber’s IP traffic to an internal in-line service:

- Step 1** Define an IP ACL as described in the [Defining IP Access Lists for Internal CSS](#) section.
- Step 2** *Optional:* Apply an ACL to an individual subscriber as described in the [Applying an ACL to an Individual Subscriber \(Optional\)](#) section.
- Step 3** *Optional:* Apply a single ACL to multiple subscribers as described in the [Applying an ACL to Multiple Subscribers \(Optional\)](#) section.
- Step 4** *Optional:* Apply an ACL to multiple subscribers via APNs as described in the [Applying an ACL to Multiple Subscribers via APNs \(Optional\)](#) section.
- Step 5** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.



Important: Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands. Not all commands or keywords/variables may be supported or available. Availability varies on the platform type and installed license(s).

Defining IP Access Lists for Internal CSS

IP ACLs specify what type of subscriber traffic and which direction (uplink, downlink, or both) traffic is redirected. The IP ACL must be specified in the context in which subscriber authentication is performed.

 **Caution:** To minimize the risk of data loss, do not make configuration changes to ACLs while the system is facilitating subscriber sessions.

Use the following configuration example to define an IP ACL for internal CSS; start in the Exec mode of the CLI:

```
configure
  context <context_name>
    ip access-list <acl_name>
      redirect css service <service_name> <keywords> <options>
    end
```

Notes:

- <service_name> must be an ACL service name.
- For information on the keywords and options available with the **redirect css service** command, see *ACL Configuration Mode Commands* in the *Command Line Interface Reference*.
- For IPv6 ACLs, the same configurations must be done in the IPv6 ACL Configuration Mode. See *IPv6 ACL Configuration Mode Commands* in the *Command Line Interface Reference*.

Applying an ACL to an Individual Subscriber (Optional)

For information on how to apply an ACL to an individual subscriber, refer to the *Applying an ACL to an Individual Subscriber* section of *Access Control Lists*.

Applying an ACL to Multiple Subscribers (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. When configured properly, the functions can be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the "default" profile.

Applying an ACL to the Subscriber Named default (Optional)

For information on how to apply an ACL to the default subscriber, refer to the *Applying an ACL to the Subscriber Named default* section in *Access Control Lists*.

Applying an ACL to Service-specified Default Subscribers (Optional)

For information on how to apply an ACL to the subscriber to be used as the “default” profile by various system services, refer to the *Applying an ACL to Service-specified Default Subscribers* section in *Access Control Lists*.

Applying an ACL to Multiple Subscribers via APNs (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

For information on how to apply an ACL to multiple subscribers via APNs, refer to the *Applying a Single ACL to Multiple Subscribers via APNs* section in *Access Control Lists*.

Appendix F

Interchassis Session Recovery

This segment describes how to configure Interchassis Session Recovery (ICSR). The product Administration Guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product Administration Guide, before using the procedures described below.

 **Important:** ICSR is a licensed Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

This segment discusses the following:

- [Overview](#)
- [ICSR Operation](#)
- [Configuring Interchassis Session Recovery \(ICSR\)](#)
- [Updating the Operating System](#)

 **Caution:** ICSR should not be configured on chassis supporting L2TP calls.

Overview

The ICSR feature provides the highest possible availability for continuous call processing without interrupting subscriber services. ICSR allows the operator to configure geographically distant gateways for redundancy purposes. In the event of a node or gateway failure, ICSR allows sessions to be transparently routed around the failure, thus maintaining the user experience. ICSR also preserves session information and state.

ICSR is implemented through the use of redundant chassis. The chassis are configured as primary and backup, with one being active and one standby. Both chassis are connected to the same AAA server. A checkpoint duration timer controls when subscriber data is sent from the active chassis to the standby chassis. If the active chassis handling the call traffic goes out of service, the standby chassis transitions to the active state and continues processing the call traffic without interrupting the subscriber session.

The chassis determine which is active through a proprietary TCP-based connection known as the Service Redundancy Protocol (SRP) link. The SRP link is used to exchange Hello messages between the primary and backup chassis and must be maintained for proper system operation.



Important: Contact your Cisco account representative to verify whether a specific service supports ICSR as an option.

Interchassis Communication

Chassis configured to support ICSR communicate using periodic Hello messages. These messages are sent by each chassis to notify the peer of its current state. The Hello message contains information about the chassis such as its configuration and priority. A dead interval is used to set a time limit for a Hello message to be received from the chassis' peer. If the standby chassis does not receive an Hello message from the active chassis within the dead interval, the standby chassis transitions to the active state. In situations where the SRP link goes out of service, a priority scheme is used to determine which chassis processes the session. The following priority scheme is used:

- route modifier
- chassis priority
- MIO/UMIO MAC address

Checkpoint Messages

Checkpoint messages are sent from the active chassis to the standby chassis. These messages are sent at specific intervals and contain all the information needed to recreate the sessions on the standby chassis, if that chassis were to become active. Once a session exceeds the checkpoint duration, checkpoint data is collected on the session.

AAA Monitor

AAA servers are monitored using the authentication probe mechanism. AAA servers are considered Up if the authentication-probe receives a valid response. AAA servers are considered Down when the **max-retries count** specified in the configuration of the AAA server has been reached. SRP initiates a switchover when none of the configured AAA servers responds to an authentication probe. AAA probing is only performed on the active chassis.



Important: A switchover event caused by an AAA monitoring failure is non-revertible.

If the newly active chassis fails to monitor the configured AAA servers, it remains as the active chassis until one of the following occurs:

- a manual switchover
- another non-AAA failure event causes the system to switchover
- a CLI command is used to clear the AAA failure flag and allow the chassis to switch to standby

BGP Interaction

The Service Redundancy Protocol implements revertible switchover behavior via a mechanism that adjusts the route modifier value for the advertised loopback/IP Pool routes. The initial value of the route modifier value is determined by the chassis' configured role and is initialized to a value that is higher than a normal operational value. This ensures that in the event of an SRP link failure and an SRP task failure, the correct chassis is still preferred in the routing domain.

The Active and Standby chassis share current route modifier values. When BGP advertises the loopback and IP pool routes, it converts the route modifier into an autonomous systems (AS) path prepend count. The Active chassis always has a lower route modifier, and thus prepends less to the AS-path attribute. This causes the route to be preferred in the routing domain.

If communication on the SRP link is lost, and both chassis in the redundant pair are claiming to be Active, the previously Active chassis is still preferred since it is advertising a smaller AS-path into the BGP routing domain. The route modifier is incremented as switchover events occur. A threshold determines when the route modifier should be reset to its initial value to avoid rollover.

Requirements

ICSR configurations require the following:

- Two chassis configured for the same service types. The services must be bound on an SRP-activated loopback interface.
- Both chassis must have identical hardware.
- Three contexts:
 - **Redundancy** – to configure the primary and backup chassis redundancy.
 - **Source** – AAA configuration of the specified nas-ip-address must be the IP address of an interface bound to an HA, or any core network service configured within the same context.
 - **Destination** – to configure monitoring and routing to the PDN.
- AAA RADIUS server
- Border Gateway Protocol (BGP) – ICSR uses the route modifier to determine the chassis priority.

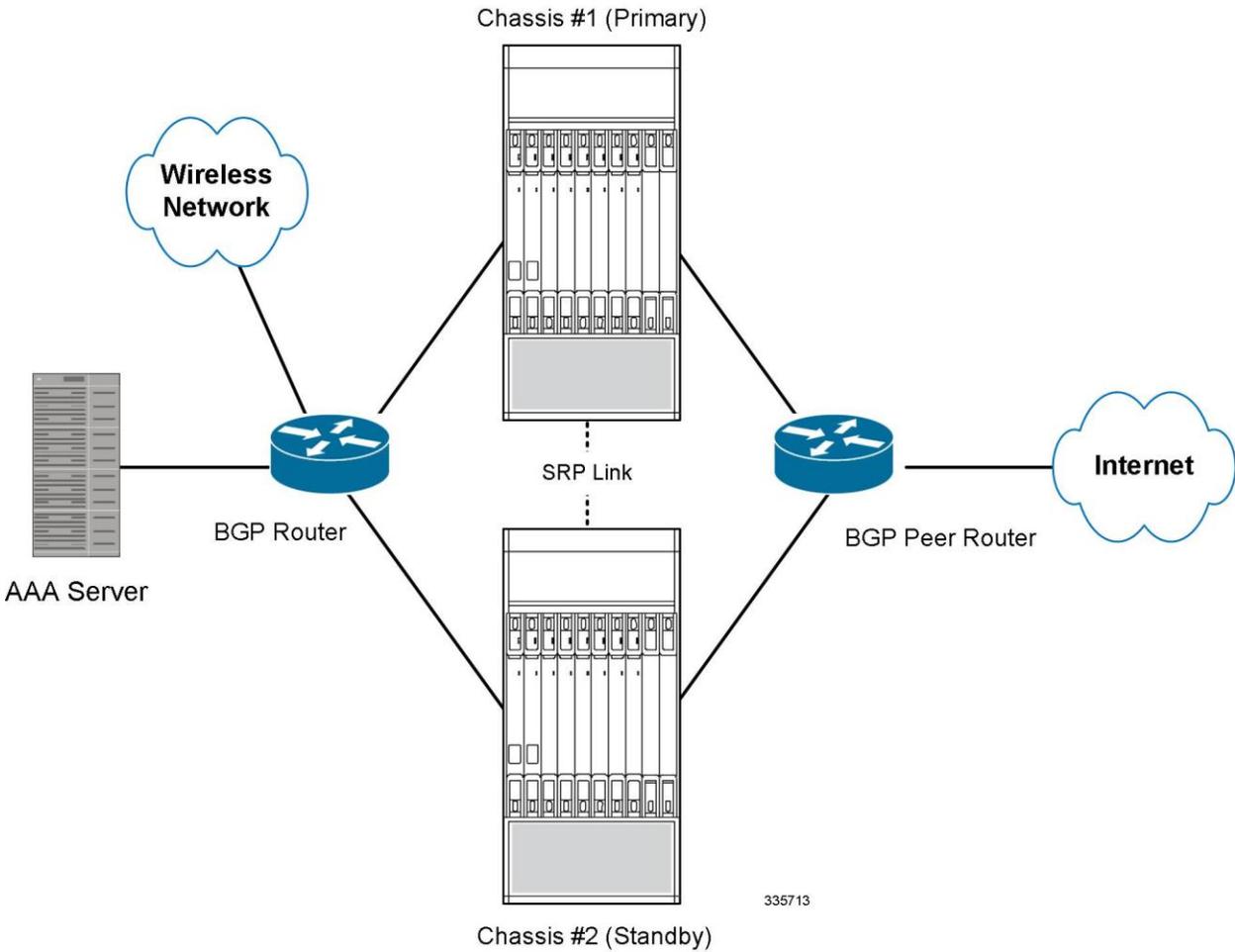


Important: ICSR is a licensed Cisco feature. Verify that each chassis has the appropriate license before using these procedures. To do this, log in to both chassis and execute a **show license information** command. Look for “Inter-Chassis Session Recovery”. If the chassis is not licensed, please contact your Cisco account representative.

Caution: ICSR should not be configured for chassis supporting L2TP calls.

The following figure shows an ICSR network.

Figure 15. ASR 5500 ICSR Network

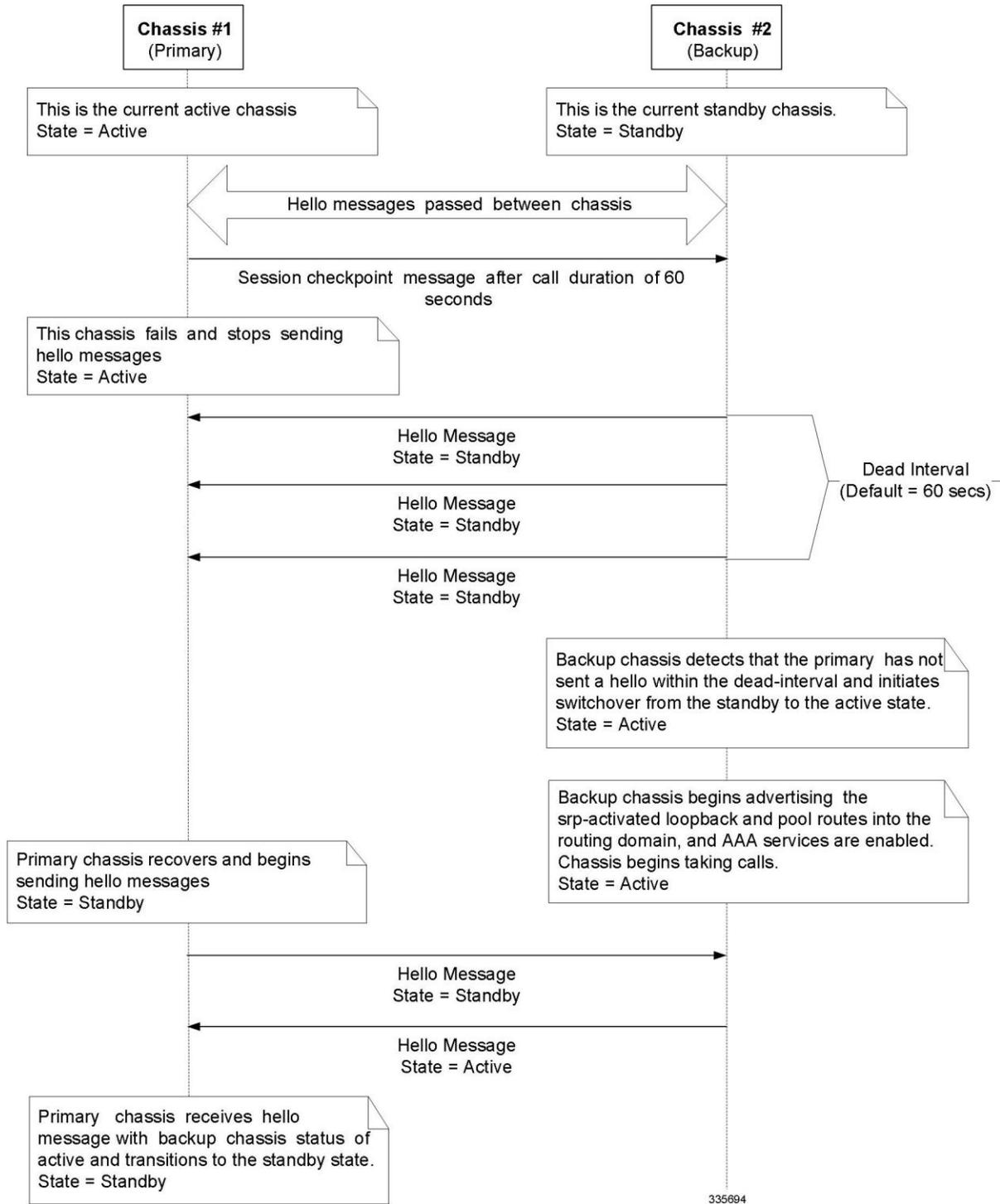


ICSR Operation

This section shows operational flows for ICSR.

The following figure shows an ICSR process flow due to a primary failure.

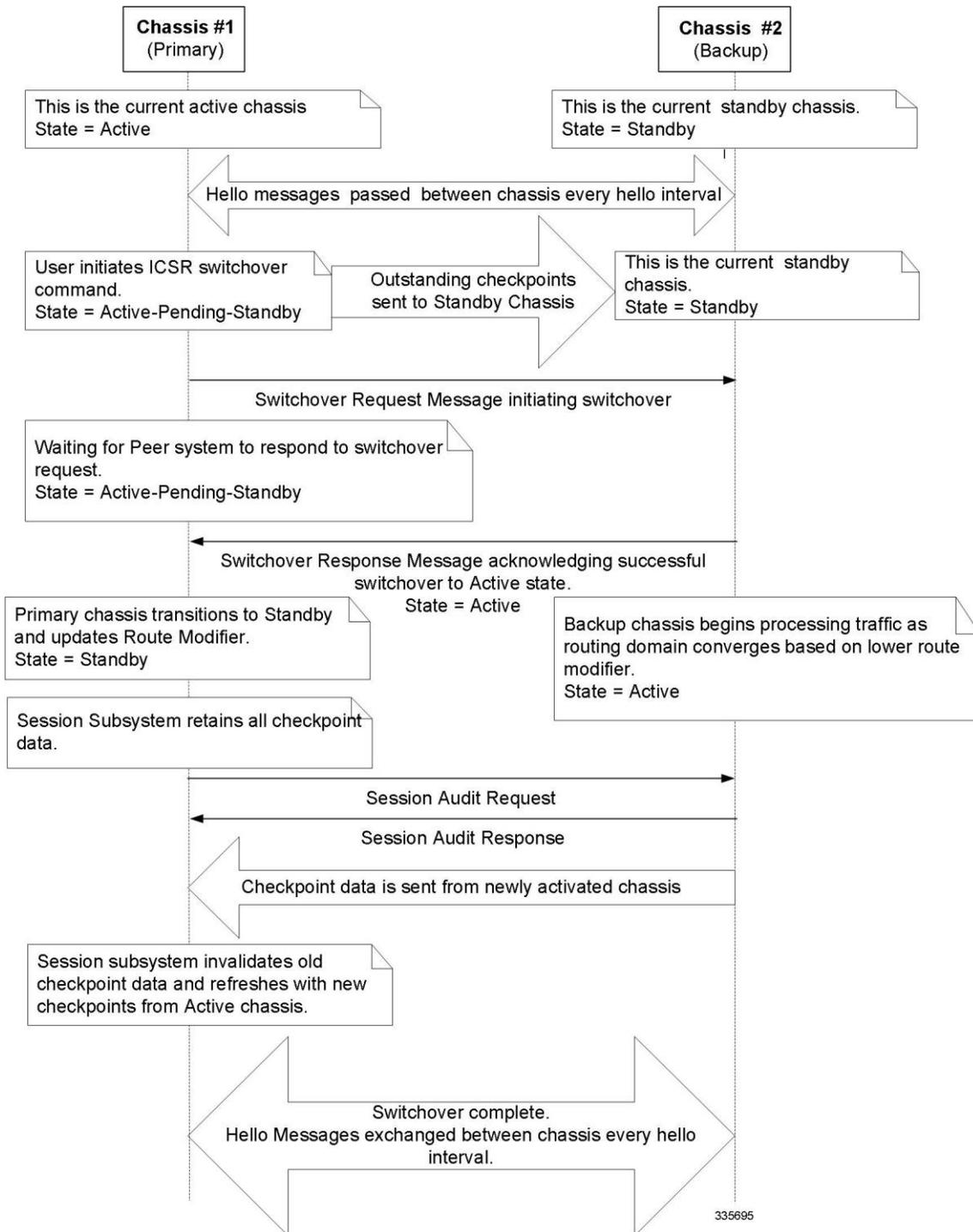
Figure 16. ICSR Process Flow (Primary Failure)



335694

The following figure shows an ICSR process flow due to a manual switchover.

Figure 17. ICSR Process Flow (Manual Switchover)



Chassis Initialization

When the chassis are simultaneously initialized, they send Hello messages to their configured peer. The peer sends a response, establishes communication between the chassis, and messages are sent that contain configuration information.

During initialization, if both chassis are misconfigured in the same mode - both active (primary) or both standby (backup), the chassis with the highest priority (lowest number set with the ICSR **priority** command) becomes active and the other chassis becomes the standby.

If the chassis priorities are the same, the system compares the two MAC addresses and the chassis with the higher MIO/UMIO MAC address becomes active. For example, if the chassis have MAC addresses of *00-02-43-03-1C-2B* and *00-02-43-03-01-3B*, the last 3 sets of octets (the first 3 sets are the vendor code) are compared. In this example, the *03-1C-2B* and *03-01-3B* are compared from left to right. The first pair of octets in both MAC addresses are the same, so the next pairs are compared. Since the *01* is lower than the *1C*, the chassis with the MIO/UMIO MAC address of *00-02-43-03-1C-2B* becomes active and the other chassis the standby.

Chassis Operation

This section describes how the chassis communicate, maintain subscriber sessions, and perform chassis switchover.

Chassis Communication

If one chassis is in the active state and one in the standby state, they both send Hello messages at each hello interval. Subscriber sessions that exceed the checkpoint session duration are included in checkpoint messages that are sent to the standby chassis. The checkpoint message contains subscriber session information so if the active chassis goes out of service, the backup chassis becomes active and is able to continue processing the subscriber sessions. Additional checkpoint messages occur at various intervals whenever subscriber session information is updated on the standby chassis.

Chassis Switchover

If the active chassis goes out of service, the standby chassis continues to send Hello messages. If the standby chassis does not receive a response to the Hello messages within the dead interval, the standby chassis initiates a switchover. During the switchover, the standby chassis begins advertising its srp-activated loopback and pool routes into the routing domain. Once the chassis becomes active, it continues to process existing AAA services and subscriber sessions that had checkpoint information, and is also able to establish new subscriber sessions.

When the primary chassis is back in service, it sends Hello messages to the configured peer. The peer sends a response, establishes communication between the chassis, and sends Hello messages that contain configuration information. The primary chassis receives an Hello message that shows the backup chassis state as active and then transitions to standby. The Hello messages continue to be sent to each peer, and checkpoint information is now sent from the active chassis to the standby chassis at regular intervals.

When chassis switchover occurs, the session timers are recovered. The access gateway session recovery is recreated with the full lifetime to avoid potential loss of the session and the possibility that a renewal update was lost in the transitional checkpoint update process.

Configuring Interchassis Session Recovery (ICSR)



Important: The ICSR configuration must be the same on the primary and backup chassis. If each chassis has a different Service Redundancy Protocol (SRP) configuration, the session recovery feature does not function and sessions cannot be recovered when the active chassis goes out of service.

This section describes how to configure basic ICSR on each chassis. For information on commands that configure additional parameters and options, refer to the *Command Line Interface Reference*.



Caution: ICSR should not be configured for chassis supporting L2TP calls.

The procedures described below assume the following:

- The chassis have been installed and configured with core network services.
For more configuration information and instructions on configuring services, refer to the respective product Administration Guide.
- In addition, the IP address pools must be **srp activated**.
- AAA server is installed, configured and accessible by both chassis.
For more information on configuring the AAA server, refer to the *AAA Interface Administration and Reference*.
- BGP router installed and configured. See *Routing* for more information on configuring BGP services.

To configure the ICSR on a primary and/or backup chassis:

- Step 1** Configure the SRP context by applying the example configuration in the [Configuring the Service Redundancy Protocol \(SRP\) Context](#) section.
- Step 2** Modify the source context of the core network service by applying the example configuration in the [Modifying the Source Context for ICSR](#) section.
- Step 3** Modify the destination context of core network service by applying the example configuration in the [Modifying the Destination Context for ICSR](#) section.
- Step 4** *Optional:* Disable bulk statistics collection on the standby system by applying the example configuration in the [Disabling Bulk Statistics Collection on a Standby System](#) section.
- Step 5** Verify your primary and backup chassis configuration by following the steps in the [Verifying the Primary and Backup Chassis Configuration](#) section.
- Step 6** Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring the Service Redundancy Protocol (SRP) Context

To configure the system to work with ICSR:

- Step 1** Create the chassis redundancy context and bind it to the IP address of the primary chassis by applying the example configuration in the [Creating and Binding the SRP Context](#) section.
- Step 2** Configure the chassis redundancy context with priority, chassis mode, hello interval, dead-interval and peer IP address by applying the example configuration in the [Configuring the SRP Context Parameters](#) section.
- Step 3** Configure the SRP context with interface parameters (including interface name, IP address and port number) for interchassis communication by applying the example configuration in the [Configuring the SRP Context Interface Parameters](#) section.
- Step 4** Verify your SRP context configuration by following the steps in the [Verifying SRP Configuration](#) section.
- Step 5** Save your configuration as described in *Verifying and Saving Your Configuration*.

Creating and Binding the SRP Context

Use the example below to create the SRP context and bind it to primary chassis IP address:

 **Important:** ICSR is configured using two systems. Be sure to create the redundancy context on both systems. CLI commands must be executed on both systems. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first. Before starting this configuration, identify which chassis to configure as the primary and use that login session.

```
configure

context <srp_ctxt_name> [-noconfirm]

    service-redundancy-protocol

        bind address <ip_address>

    end
```

Notes:

- ICSR should be configured and maintained in a separate context.
- Be sure to bind the local IP address to the primary chassis. When configuring the backup chassis, be sure to bind the local IP address to the backup chassis.

Configuring the SRP Context Parameters

This configuration assigns a chassis mode and priority, and also configures the redundancy link between the primary and backup chassis:

 **Important:** CLI commands must be executed on both chassis. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first.

```

configure

  context <srp_ctxt_name>

    service-redundancy-protocol

      chassis-mode { primary | backup }

      priority <priority>

      peer-ip-address <ip_address>

      hello-interval <dur_sec>

      dead-interval <dead_dur_sec>

    end

```

Notes:

- ICSR should be configured and maintained in a separate context.
- When assigning the chassis mode on the backup chassis be sure to enter the **backup** keyword.
- The **priority** determines which chassis becomes active when the redundancy link goes out of service. The higher priority chassis has the lower number. Be sure to assign different priorities to each chassis.
- Enter the IP chassis of the backup chassis as the **peer-ip-address** to the primary chassis. Assign the IP address of the primary chassis as the **peer-ip-address** to the backup chassis.
- The **dead-interval** must be at least three times greater than the **hello-interval**. For example, if the hello interval is 10, the dead interval should be at least 30. System performance is severely impacted if the hello interval and dead interval are not set properly.

Configuring the SRP Context Interface Parameters

This procedure configures the communication interface with the IP address and port number within the SRP context. This interface supports interchassis communication.

 **Important:** CLI commands must be executed on both chassis. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first.

```

configure

  context <vpn_ctxt_name> [-noconfirm]

    interface <srp_if_name>

      ip-address { <ip_address> | <ip_address>/<mask> }

    exit

```

```

    exit

port ethernet <slot_num>/<port_num>

    description <des_string>

    medium { auto | speed { 10 | 100 | 1000 } duplex { full | half } }

    no shutdown

    bind interface <srp_if_name> <srp_ctxt_name>

end

```

Verifying SRP Configuration

Step 1 Verify that your SRP contexts were created and configured properly by entering the **show srp info** command (Exec Mode).

Sample output for this command as shown. In this example, an SRP context called *srp1* was configured with default parameters.

```

Service Redundancy Protocol:
-----
Context:                srp1
Local Address:          0.0.0.0
Chassis State:          Init
Chassis Mode:           Backup
Chassis Priority:        125
Local Tiebreaker:       00-00-00-00-00-00
Route-Modifier:         34
Peer Remote Address:    0.0.0.0
Peer State:              Init
Peer Mode:               Init
Peer Priority:           0
Peer Tiebreaker:        00-00-00-00-00-00
Peer Route-Modifier:    0
Last Hello Message received: -
Peer Configuration Validation: Initial
Last Peer Configuration Error: None
Last Peer Configuration Event: -
Connection State:       None

```

Modifying the Source Context for ICSR

To modify the source context of core service:

- Step 1** Add the Border Gateway Protocol (BGP) router AS-path and configure the gateway IP address, neighbor IP address, remote IP address in the source context where the core network service is configured, by applying the example configuration in the [Configuring BGP Router and Gateway Address](#) section.
- Step 2** Configure the service redundancy context with the BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in the [Configuring SRP Context for BGP](#) section.
- Step 3** Verify your BGP context configuration by following the steps in the [Verifying BGP Configuration](#) section.
- Step 4** Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring BGP Router and Gateway Address

Use the following example to create the BGP context and network addresses.

```
configure
  context <source_ctxt_name>
    router bgp <AS_num>
      network <gw_ip_address>
      neighbor <neighbor_ip_address> remote-as <AS_num>
    end
```

Notes:

- *source_ctxt_name* is the context where the core network service is configured.

Configuring the SRP Context for BGP

Use the following example to configure the BGP context and IP addresses in the SRP context.

```
configure
  context <srp_ctxt_name>
    service-redundancy-protocol
      monitor bgp context <source_ctxt_name> <neighbor_ip_address>
    end
```

neighbor_ip_address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. Multiple IP addresses can be added per context as IPv4 and/or IPv6 IP addresses.

By default an ICSR failover is triggered when all BGP peers within a context are down.

Verifying BGP Configuration

Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Modifying the Destination Context for ICSR

To modify the destination context of core service:

- Step 1** Add the BGP router and configure the gateway IP address, neighbor IP address, remote IP address in the destination context where the core network service is configured, by applying the example configuration in the [Configuring BGP Router and Gateway Address in Destination Context](#) section.
- Step 2** Configure the service redundancy context with BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in the [Configuring SRP Context for BGP for Destination Context](#) section.
- Step 3** Set the subscriber mode to **default** by following the steps in the [Setting Subscriber to Default Mode](#) section.
- Step 4** Verify your BGP context configuration by following the steps in the [Verifying BGP Configuration in Destination Context](#) section.
- Step 5** Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring BGP Router and Gateway Address in Destination Context

Use the following example to create the BGP context and network addresses.

```

configure

  context <dest_ctxt_name>

    router bgp <AS_num>

      network <gw_ip_address>

      neighbor <neighbor_ip_address> remote-as <AS_num>

    end

```

Notes:

- *AS_num* is the autonomous systems path number for this BGP router.

Configuring SRP Context for BGP for Destination Context

Use the following example to configure the BGP context and IP addresses in the SRP context.

```

configure

  context <srp_ctxt_name>

    service-redundancy-protocol

      monitor bgp context <dest_ctxt_name> <neighbor_ip_address>

    end

```

Setting Subscriber to Default Mode

Use the following example to set the subscriber mode to **default**.

```
configure
  context <dest_ctxt_name>
    subscriber default
  end
```

Verifying BGP Configuration in Destination Context

Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Disabling Bulk Statistics Collection on a Standby System

You can disable the collection of bulk statistics from a system when it is in the standby mode of operation.

 **Important:** When this feature is enabled and a system transitions to standby state, any pending accumulated statistical data is transferred at the first opportunity. After that no additional statistics gathering takes place until the system comes out of standby state.

Use the following example to disable the bulk statistics collection on a standby system.

```
configure
  bulkstat mode
    no gather-on-standby
  end
```

Repeat this procedure for both systems.

Verifying the Primary and Backup Chassis Configuration

This section describes how to compare the ICSR configuration on both chassis.

Step 1 Enter the **show configuration srp** command on both chassis (Exec mode).

Verify that both chassis have the same SRP configuration information. The output looks similar to following:

```
config
  context source
    interface haservice loopback
      ip address 172.17.1.1 255.255.255.255 srp-activate
    #exit
    radius attribute nas-ip-address address 172.17.1.1
    radius server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1812
```

```
radius accounting server 192.168.83.2 encrypted key 01abd002c82b4a2c port
1813
ha-service ha-pdsn
mn-ha-spi spi-number 256 encrypted secret 6c93f7960b726b6f6c93f7960b726b6f
hash-algorithm md5
fa-ha-spi remote-address 192.168.82.0/24 spi-number 256 encrypted secret
1088bdd6817f64df
bind address 172.17.1.1
#exit
#exit
context destination
ip pool dynamic 172.18.0.0 255.255.0.0 public 0 srp-activate
ip pool static 172.19.0.0 255.255.240.0 static srp-activate
#exit
context srp
service-redundancy-protocol
#exit
#exit
end
```

Updating the Operating System

Updating the operating system (StarOS™) on ICSR chassis requires performing an Off-line update of each chassis while it is standby mode. Traffic disruption is minimal since an active chassis will be handling call sessions while the standby chassis is being updated.

The general upgrade sequence is as follows:

- Download the StarOS software image and copy/transfer it to both chassis.
- Save the currently running configurations on both chassis.
- Update the standby backup chassis first.
- Initiate an SRP switchover from the active primary chassis to make the standby backup chassis active.
- Update the standby primary chassis.
- Initiate an SRP switchover from the active backup chassis to make the standby primary chassis active.

The four-part flowchart below shows a more complete view of all the procedures required to complete the StarOS upgrade process.

Figure 18. ICSR Software Upgrade – Part 1



Figure 19. ICSR Software Upgrade – Part 2

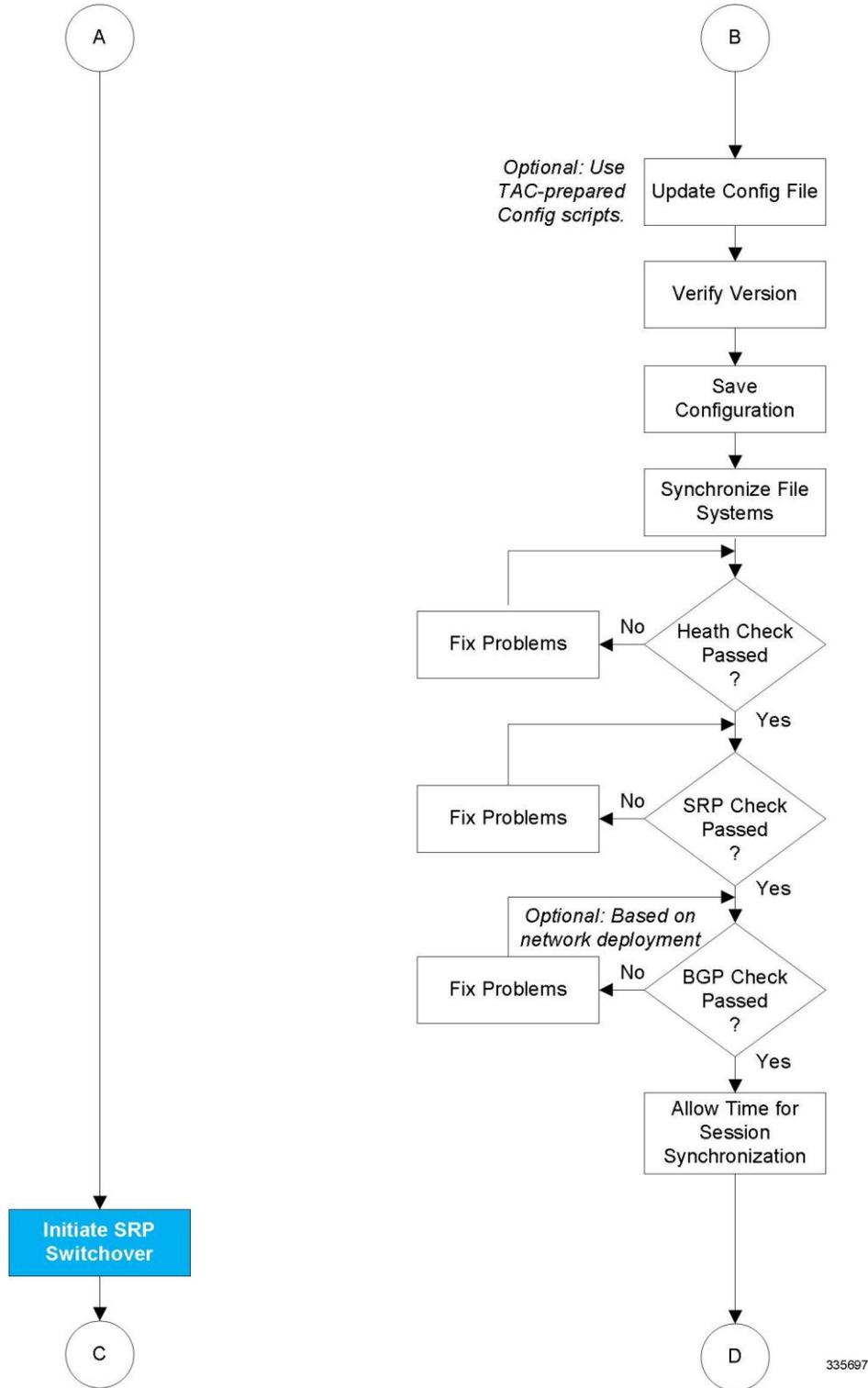
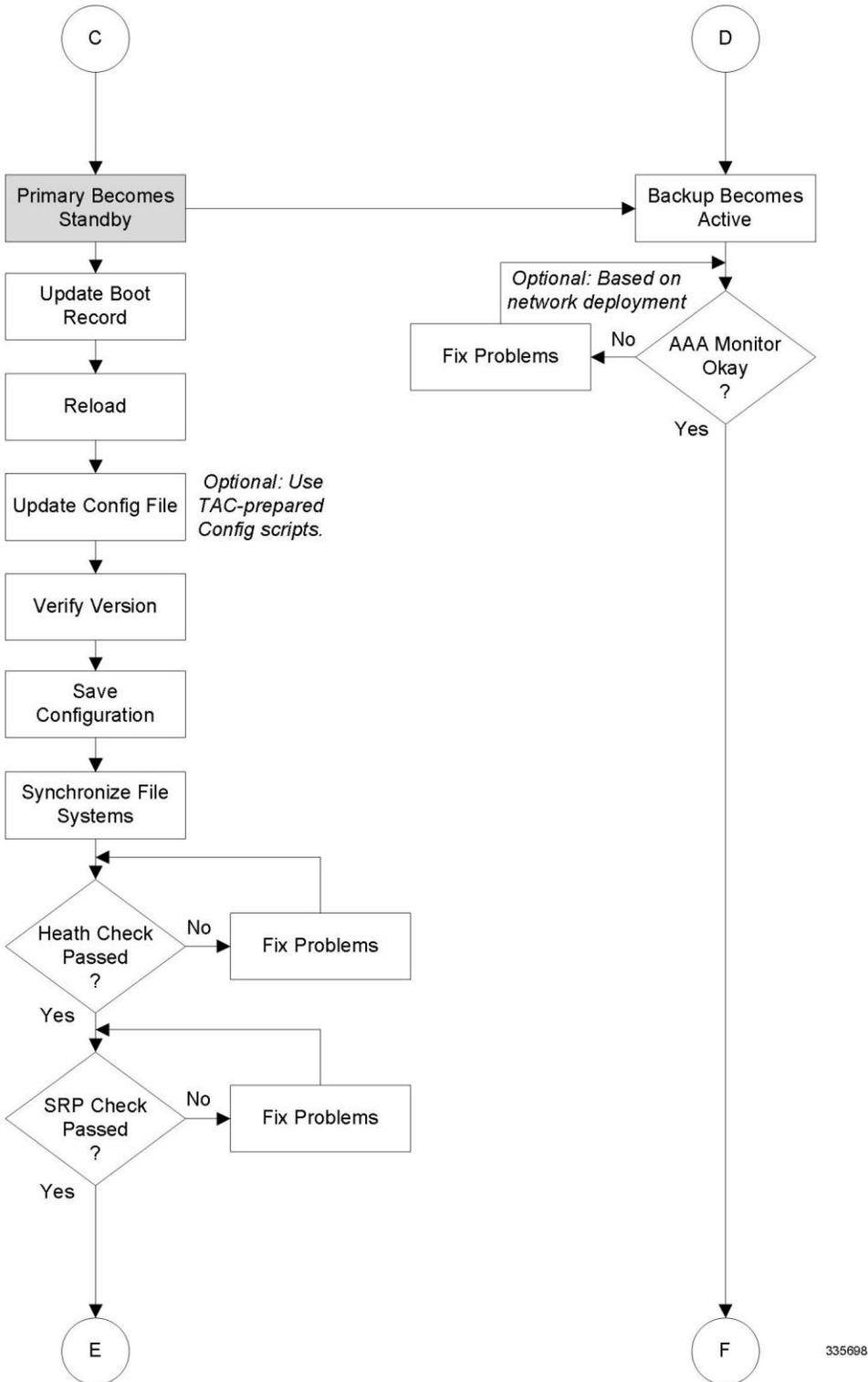
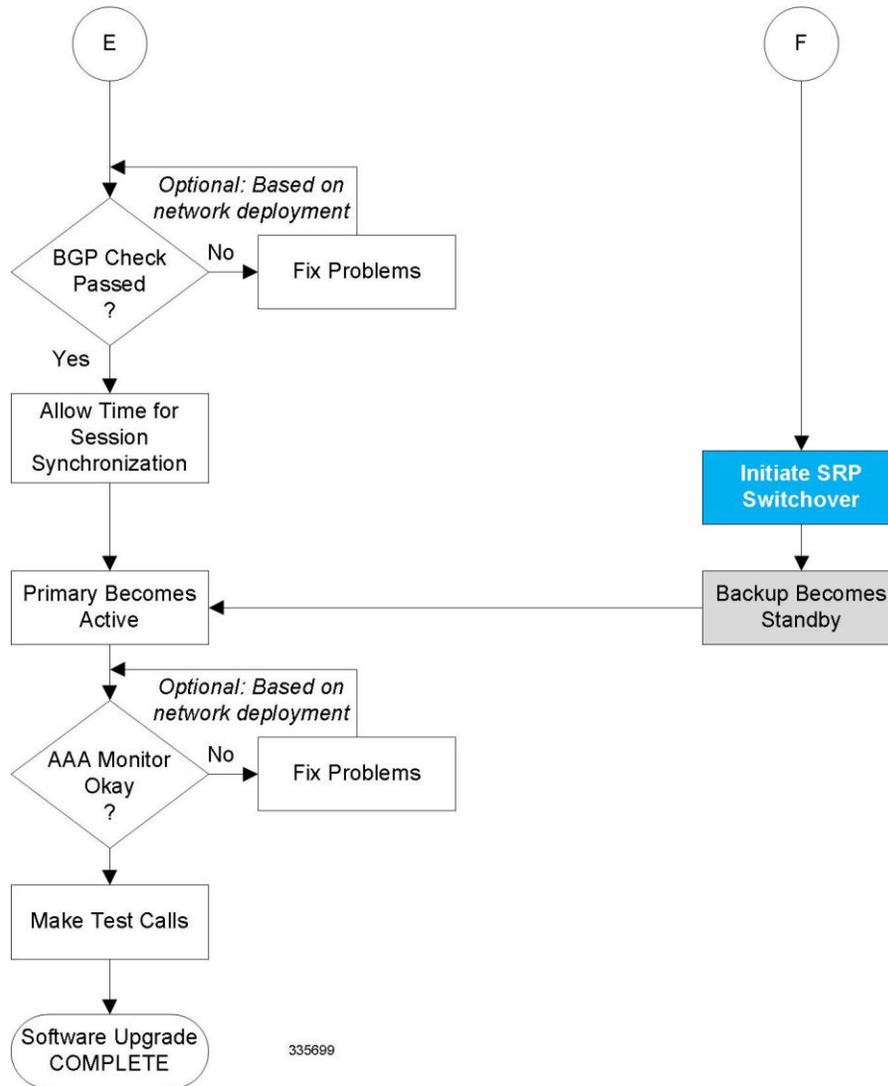


Figure 20. ICSR Software Upgrade – Part 3



335698

Figure 21. ICSR Software Upgrade – Part 4



Both ICSR Chassis

Log into the CLI of the primary and backup and perform the tasks described below.

Downloading and Transferring the StarOS Build

Step 1 Verify that there is enough free space on the **/flash** device to accommodate the new operating system image file by entering the following Exec mode command:

```
[local]<host_name># directory /flash
```

Step 2 Access to the Cisco support site and download facility is username and password controlled. Download the software image to a network location or physical device (USB stick) from which it can be uploaded to the **/flash** device.

Step 3 Transfer the new operating system image file to the **/flash** device on the MIO/UMIO using one of the following methods:

Step a Copy the file from a network location or local device plugged into the MIO/UMIO using the **copy** command

```
[local]<host_name># copy <from_url> <to_url> [-noconfirm]
```

Step b Transfer the file to the **/flash** device using an FTP client with access to the system. The FTP client must be configured to transfer the file using binary mode.

Step c Transfer the file to the **/flash** device using an SFTP client with access to the system.

Step 4 Verify that the image file was successfully transferred to the **/flash** device by running the Exec mode the following command

```
[local]<host_name># directory /flash
```

Step 5 Run the **show version /flash/<image_filename>** command to verify the build information. For example:

```
local]<host_name># show version /flash/production.43582.asr5500.bin
```

Standby Backup Chassis

Log into the backup standby chassis and perform the tasks described below.

Performing Health Checks

Health checks are a series of Exec mode **show** commands to determine the readiness of the system to handle a software update.

- Step 1** Run **show card table all |grep unknown**. No output should be displayed.
- Step 2** Run **show card table |grep offline**. No output should be displayed.
- Step 3** Run **show resources |grep Status**. The output should display “within acceptable limits”.
- Step 4** Run **show alarm outstanding**. Review the output for any issues that may preclude performing the software update.

Performing SRP Checks

Service Redundancy Protocol (SRP) checks verify that the mechanism for monitoring ICSR system status is operational.

- Step 1** Run **show srp monitor all**.
- Step 2** Review the output for any issues that may preclude performing the software update.

Performing BGP Checks

Border Gateway Protocol (BGP) checks are only required when BGP is used to support redundant interchassis communication. These checks are run per context and per service type.

- Step 1** For each BGP-enabled context, run **show ip bgp summary**. Verify that the BGP peers are connected and IPv4 and IPv6 peers are up. Repeat for all BGP-enable contexts.
- Step 2** Run **show <service_name> all |grep "Service Status:"**. The service should be “Started”. Repeat for all services running on the chassis.

Updating the Boot Record

You must add a new boot stack entry for the recently downloaded software image (.bin) file.

- Step 1** Run the Exec mode **show boot** command to verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).
- Step 2** Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:

```
[local]<host_name>(config)# boot system priority <number> image <image_url>  
/flash/<filename> config <cfg_url> /flash/<filename>
```

Step 3 Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the **no boot system priority** command to delete a boot stack entry.

For information on using the **boot system priority** command, refer to the *Adding a New Boot Stack Entry* section in this guide.

Synchronizing File Systems

Synchronize the local file systems by entering the following Exec mode command:

```
[local]<host_name># filesystem synchronize all
```

Reloading the Chassis

Reboot the chassis by entering the following command:

```
[local]<host_name># reload [-noconfirm]
```

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

After the system reboots, establish a CLI session and enter the **show version** command to verify that the active software version is correct.

Optional for PDSN: If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

Updating the Configuration File

Features in the new operating system may require changes to the configuration file. These changes can be done manually or facilitated by custom scripts prepared by Cisco TAC. Make whatever changes are necessary prior to saving the updated configuration file.

Verifying the Software Version

After the system has successfully booted, verify that the new StarOS version is running by executing the Exec mode **show version** command.

Saving the Configuration File

Use the Exec mode save configuration command to save the currently running configuration to the **/flash** device and to an off-chassis location (external memory device or network URL). The off-chassis copy assures that you will have a fallback, loadable configuration file should a problem be encountered.

Completing the Update Process

Repeat the following tasks to complete the upgrade process on the standby secondary chassis:

- [Synchronizing File Systems](#)
- [Performing Health Checks](#)
- [Performing SRP Checks](#)
- [Performing BGP Checks](#)

Waiting for Session Synchronization

Allow time for session synchronization to occur between the ICSR chassis before proceeding to the next steps.

- Step 1** Run the **show session recovery status verbose** command on both chassis. Proceed to the next steps only when no errors are seen in the output of this command.
- Step 2** On the standby chassis, run **show srp checkpoint statistics |more**.
- Step 3** On active chassis, run **show subs summary |grep Total**.
- Step 4** Compare the number of subscribers on the active chassis and the number of "Current pre-allocated calls:" on the standby chassis. They should be similar (within 5%). Allow a few minutes for systems to complete synchronization.

Primary Chassis

Log into the active primary chassis and complete the tasks described below.

Initiating an SRP Switchover

An SRP switchover places the primary chassis in standby mode and makes the backup chassis active. The secondary chassis is now processing sessions with the upgraded software.

- Step 1** On the primary chassis run the **srp initiate-switchover** command. All existing sessions will be migrated to the backup chassis and it begins servicing new session requests. Allow the switchover process to complete.
- Step 2** On the primary chassis, run the **show srp info** command. Chassis State should indicate Standby when switchover is complete.
- Step 3** On the backup chassis, confirm the switchover is complete by running the **show srp info** command. Chassis State should indicate Active when switchover is complete.

Checking AAA Monitor Status on the Newly Active Chassis

If your network deployment requires communication with AAA servers, log into the newly active chassis and perform an AAA monitor check. You will be checking for the existence of any SNMP traps that indicate the chassis cannot communicate with AAA servers (**starSRPAAAUnreachable**).

- Step 1** Run the Exec mode command **show snmp trap history |grep starSRPAAAUnreachable**.
- Step 2** There should be no output for this command, or no very recent SNMP trap notifications (based on the event timestamp).
- Step 3** If the active chassis cannot communicate with one or more AAA servers, refer to the [AAA Monitor](#) section for additional information on how to proceed.

Completing the Software Update

Log into the standby chassis and repeat the following tasks to complete the upgrade process on the standby primary chassis:

- [Updating the Boot Record](#)
- [Reloading the Chassis](#)
- [Updating the Configuration File](#)
- [Verifying the Software Version](#)
- [Saving the Configuration File](#)
- [Synchronizing File Systems](#)
- [Performing Health Checks](#)
- [Performing SRP Checks](#)
- [Performing BGP Checks](#)
- [Waiting for Session Synchronization](#)

Initiating an SRP Switchover

An SRP switchover places the primary chassis in active mode and makes the backup chassis active. The primary chassis is now processing sessions with the upgraded software.

- Step 1** On the backup chassis run the **srp initiate-switchover** command. All existing sessions will be migrated to the primary chassis and it begins servicing new session requests. Allow the switchover process to complete.
- Step 2** On the backup chassis, run the **show srp info** command. Chassis State should indicate Standby when switchover is complete.
- Step 3** On the primary chassis, confirm the switchover is complete by running the **show srp info** command. Chassis State should indicate Active when switchover is complete.

Checking AAA Monitor Status

If your network deployment requires communication with AAA servers, check the status of communication with AAA servers as described in the [Checking AAA Monitor Status on the Newly Active Chassis](#) section.

Making Test Calls

Once the chassis state is verified and subscribers are migrated, perform new call testing to make sure calls are successful.

Fallback Procedure

To revert to the previous configuration and software build, perform the following steps as a user with administrative privileges.

- Step 1** Run the Exec mode **show boot** command. The topmost lowest numbered entry of the displayed output should be the new configuration with the new software build. The second topmost entry should be the backup configuration.
- Step 2** Remove the topmost boot entry <n>, and synchronize the configuration across the management cards

```
[local]<host_name># config

[local]<host_name>(config)# no boot system priority <n>

[local]<host_name>(config)# end

[local]<host_name># filesystem synchronize all
```

- Step 3** Reboot the system to load its previous configuration.

```
[local]<host_name># reload
```

- Step 4** Perform health checks as described in [Performing Health Checks](#).

Appendix G

Routing

This segment provides information on configuring an enhanced, or extended, service. The product administration guides provide examples and procedures for configuring basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model before using the procedures idescribed below.

This segment includes the following sections:

- [Routing Policies](#)
- [Static Routing](#)
- [OSPF Routing](#)
- [OSPFv3 Routing](#)
- [Viewing Routing Information](#)
- [Equal Cost Multiple Path \(ECMP\)](#)
- [BGP-4 Routing](#)
- [Bidirectional Forwarding Detection](#)

Routing Policies

This section describes how to configure the elements needed to define routing policies. Routing policies modify and redirect routes to and from the system to satisfy specific network deployment requirements.

Use the following building blocks to configure routing policies:

- **Route Access Lists** – The basic building block of a routing policy. Route access lists filter routes based on a range of IP addresses.
- **IP Prefix Lists** – A more advanced element of a routing policy. An IP Prefix list filters routes based on IP prefixes.
- **AS Path Access Lists** – A basic building block used for Border Gateway Protocol (BGP) routing. These lists filter Autonomous System (AS) paths.
- **Route Maps** – Route-maps provide detailed control over routes during route selection or route advertisement by a routing protocol, and in route redistribution between routing protocols. For this level of control you use IP Prefix Lists, Route Access Lists and AS Path Access Lists to specify IP addresses, address ranges, and Autonomous System paths.

Creating IP Prefix Lists

Use the following configuration example to create IP Prefix Lists:

```
config
    context <context_name>
        ip prefix-list name <list_name> { deny | permit }
        <network_address/net_mask>
```

Notes:

- Set the IP prefix list to deny, permit or match any prefix.
- IPv4 dotted-decimal and IPv6 colon-separated-hexadecimal addresses are supported.
- Save your configuration as described in *Verifying and Saving Your Configuration*.

Creating Route Access Lists

Use the following procedure to create a Route Access List:

```
config
    context <context_name>
        route-access-list { extended identifier } { deny | permit } [ ip address
        <ip_address> ]
        route-access-list named <list_name> { deny | permit } { <ip_address/mask>
        | any } [ exact-match ]
```

```
route-access-list standard <identifier> { permit | deny } { <ip_address>
<wildcard_mask> | any | <network_address> }
```

Notes:

- A maximum of 64 access lists are supported per context.
- Save your configuration as described in *Verifying and Saving Your Configuration*.

Creating AS Path Access Lists

Use the following procedure to create an AS Path Access List:

```
config
    context <context_name>
        ip as-path access-list <list_name> [ { deny | permit } <reg_expr> ]
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Creating Route Maps

Use the following configuration example to create a Route Map:

```
config
    context <context_name>
        route-map <map_name> { deny | permit } <seq_number>
```

Notes:

- Use the **match** and **set** commands in Route Map Configuration mode to configure the route map. Refer to the *Command Line Interface Reference* for more information on these commands.
- Save your configuration as described in *Verifying and Saving Your Configuration*.

Sample Configuration

The example below shows a configuration that creates two route access lists, applies them to a route map, and uses that route map for a BGP router neighbor.

```
config
  context ispl
    route-access-list named RACLin1a permit 88.151.1.0/30
    route-access-list named RACLin1a permit 88.151.1.4/30
    route-access-list named RACLany permit any
    route-map RMnet1 deny 100
      match ip address route-access-list RACLin 1 a
      #exit
    route-map RMnet1 deny 200
      match ip address route-access-list RACLin 1 b
      #exit
    route-map RMnet1 permit 1000
      match ip address route-access-list RACLany
      #exit
  router bgp 1
    neighbor 152.20.1.99 as-path 101
    neighbor 152.20.1.99 route-map RMnet1
```

Static Routing

The system supports static network route configuration on a per context basis. Define network routes by specifying the:

- IP address and mask for the route
- Name of the interface in the current context that the route must use
- Next hop IP address

Adding Static Routes to a Context

To add static routes to a context configuration, you must know the names of the interfaces that are configured in the current context. Use the **show ip interface** command to list the interfaces in the current context (Exec mode).

Information for all interfaces configured in the current context is displayed as shown in the following example.

```
[local]<host_name># show ip interface
Intf Name: Egress 1
Description:
IP State: Up (Bound to 24/1 untagged ifIndex 402718721)
IP State: Up (Bound to 5/1 untagged ifIndex 83951617)
IP Address: 192.168.231.5
Subnet Mask: 255.255.255.0
Bcast Address: 192.168.231.255
MTU: 1500
Resoln Type: ARP      ARP timeout: 3600 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
Total interface count: 1
```

The first line of information for each interface lists the interface name for the current context as shown in the example output. In this example, there is one interface with the name *Egress 1*.

config

```
context <context_name>

    ip route { <ip_address> [ <ip_mask> ] | <ip_addr_mask_combo> } { next-hop
<next_hop_address> | <egress_name> [ precedence <precedence> [ cost <cost> ]
```

Notes:

You can configure a maximum of 1,200 static routes per context. Save your configuration as described in *Verifying and Saving Your Configuration*.

Deleting Static Routes From a Context

Use the following configuration example to remove static routes from a context's configuration:

```
config
    context context_name
        no ip route { <ip_address> <ip_mask> | <ip_addr_mask_combo> }
        <next_hop_address> <egress_name> [ precedence <precedence> ] [ cost <cost> ]
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

OSPF Routing

This section gives an overview of Open Shortest Path First (OSPF) routing and its implementation in the system. It also describes how to enable the base OSPF functionality and lists the commands that are available for more complex configurations.

You must purchase and install a license key before you can use this feature. Contact your Cisco account representative for more information on licenses.



Important: During system task recovery, it is possible for a dynamically-learned forwarding entry to incorrectly remain in the system forwarding table if that forwarding entry has been removed from the dynamic routing protocol during the recovery.

OSPF Version 2 Overview

OSPF is a link-state routing protocol that employs an interior gateway protocol (IGP) to route IP packets using the shortest path first based solely on the destination IP address in the IP packet header. OSPF routed IP packets are not encapsulated in any additional protocol headers as they transit the network.

An Autonomous System (AS), or Domain, is defined as a group of networks within a common routing infrastructure.

OSPF is a dynamic routing protocol that quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence. This period of convergence is short and involves a minimum of routing traffic.

In a link-state routing protocol, each router maintains a database, referred to as the link-state database, that describes the Autonomous System's topology. Each participating router has an identical database. Each entry in this database is a particular router's local state (for example, the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the AS by flooding.

All routers run the same algorithm in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as root to each destination in the AS. Externally derived routing information appears on the tree as leaves. The cost of a route is described by a single dimensionless metric.

OSPF allows sets of networks to be grouped together. Such a grouping is called an area. The topology of this area is hidden from the rest of the AS, which enables a significant reduction in routing traffic. Also, routing within the area is determined only by the area's own topology, lending the area protection from bad routing data. An area is a generalization of an IP subnetted network.

OSPF enables the flexible configuration of IP subnets so that each route distributed by OSPF has a destination and mask. Two different subnets of the same IP network number may have different sizes (that is, different masks). This is commonly referred to as variable-length subnetting. A packet is routed to the best (longest or most specific) match. Host routes are considered to be subnets whose masks are "all ones" (0xffffffff).

OSPF traffic can be authenticated or non-authenticated, or can use no authentication, simple/clear text passwords, or MD5-based passwords. This means that only trusted routers can participate in the AS routing. You can specify a variety of authentication schemes and, in fact, you can configure separate authentication schemes for each IP subnet.

Externally derived routing data (for example, routes learned from an exterior protocol such as BGP) is advertised throughout the AS. This externally derived data is kept separate from the OSPF link state data.

Each external route can also be tagged by the advertising router, enabling the passing of additional information between routers on the boundary of the AS.

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations.

Basic OSPFv2 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPF Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
    context <context_name>
        router ospf
    end
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Enabling OSPF Over a Specific Interface

After you enable OSPF, specify the networks on which it will run. Use the following command to enable OSPF:

```
network network_ip_address/network_mask area { area_id | area_ip_address }
```

 **Important:** The default cost for OSPF on the system is 10. To change the cost, refer to the `ip ospf cost` command in the *Ethernet Interface Configuration Commands* section of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Redistributing Routes Into OSPF (Optional)

Redistributing routes into OSPF means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv2 protocol to all OSPF areas. This is an optional configuration.

```
config
    context <context_name>
        router ospf
            redistribute { connected | static }
        end
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Confirming OSPF Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ospf** in the screen output:

```
show config context ctxt_name [ verbose ]
```

OSPFv3 Routing

This section gives an overview of Open Shortest Path First Version 3 (OSPFv3) routing and its implementation in the system. It also describes how to enable the base OSPFv3 functionality and lists the commands that are available for more complex configurations.

OSPFv3 Overview

Much of OSPF version 3 is the same as OSPF version 2. OSPFv3 expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. OSPFv3 dynamically learns and advertises (redistributes) IPv6 routes within an OSPFv3 routing domain.

In OSPFv3, a routing process does not need to be explicitly created. Enabling OSPFv3 on an interface will cause a routing process and its associated configuration to be created.

Basic OSPFv3 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPFv3 Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
  context <context_name>
    router ospfv3
  end
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Enabling OSPFv6 Over a Specific Interface

After you enable OSPFv3 specify the area in which it will run. Use the following command to enable OSPFv3:

```
area { area_id | area_ip_address } [ default-cost dflt-cost ] [ stub stub-area ]
[ virtual-link vl-neighbor-ipv4address ]
```



Important: The default cost for OSPFv3 on the system is 10. To change the cost, refer to the **ipv6 ospf cost** command in the *Ethernet Interface Configuration Commands* section of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Redistributing Routes Into OSPFv3 (Optional)

Redistributing routes into OSPFv3 means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv3 protocol to all OSPF areas. This is an optional configuration.

```
config
  context <context_name>
    router ospf3
      redistribute { connected | static }
    end
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

Confirming OSPFv3 Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ipv6 ospf** in the screen output:

```
show config context ctxt_name [ verbose ]
```

Viewing Routing Information

To view routing information for the current context, run one of the following Exec mode commands;

- **show ip route**: Displays information for all types of routes in the current contexts routing table.
- **show ip static-route**: Displays information only for static routes in the current contexts routing table.
- **show ip ospf**: Displays OSPF process summary information in the current context.
- **show ipv6 ospf**: Displays OSPFv3 process summary information in the current context.

This example shows sample output of the command, **show ip route**.

```
[local]host_name# show ip route
"" indicates the Best or Used route.
```

Destination	Nexthop	Protocol	Prec	Cost	Interface
*44.44.44.0/24	208.230.231.50	static	1	0	local1
*192.168.82.0/24	0.0.0.0	connected	0	0	
*192.168.83.0/24	0.0.0.0	connected	0	0	
208.230.231.0/24	0.0.0.0	ospf	110	10	local1
*208.230.231.0/24	0.0.0.0	connected	0	0	local1

Total route count: 5

Equal Cost Multiple Path (ECMP)

The system supports ECMP for routing protocols. ECMP distributes traffic across multiple routes that have the same cost to lessen the burden on any one route.

```
config
context <context_name>
ip routing maximum-paths [ max_num ]
```

Notes:

- Save your configuration as described in *Verifying and Saving Your Configuration*.

BGP-4 Routing

The Border Gateway Protocol 4 (BGP-4) routing protocol is supported through a BGP router process that is implemented at the context level.

The Border Gateway Protocol (BGP) is an inter-AS routing protocol. An Autonomous System (AS) is a set of routers under a single technical administration that use an interior gateway protocol and common metrics to route packets within the AS. The set of routers uses an exterior gateway protocol to route packets to other autonomous systems.

BGP runs over TCP. This eliminates the need for the BGP protocol to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing information. Any authentication scheme used by TCP may be used in addition to BGP's own authentication mechanisms.

BGP routers exchange network reachability information with other BGP routers. This information builds a picture of AS connectivity from which routes are filtered and AS-level policy decisions are enforced.

BGP-4 provides classless inter-domain routing. This includes support for advertising an IP prefix and eliminates the concept of network class within BGP. BGP-4 also allows the aggregation of routes, including the aggregation of AS paths.

Overview of BGP Support

Mobile devices communicate to the Internet through Home Agents (HAs). HAs assign IP addresses to the mobile node from a configured pool of addresses. These addresses are also advertised to Internet routers through an IP routing protocol to ensure dynamic routing. The BGP-4 protocol is used as a monitoring mechanism between an HA and Internet router with routing to support Interchassis Session Recovery (ICSR). (Refer to *Interchassis Session Recovery* for more information.)

The objective of BGP-4 protocol support is to satisfy routing requirements and monitor communications with Internet routers. BGP-4 may trigger an active to standby switchover to keep subscriber services from being interrupted.

The following BGP-4 features are supported:

- Exterior Border Gateway Protocol (EBGP) multi-hop
- Route Filtering for inbound and outbound routes
- Route redistribution and route-maps

IP pool routes and loopback routes are advertised in the BGP domain in the following ways:

- Through BGP Configuration Mode **redistribution** commands, all or some of the connected routes are redistributed into the BGP domain. (IP pool and loopback routes are present in the IP routing table as connected routes.) The **network routemap** command provides the flexibility to change many BGP attributes.
- Through the BGP Configuration Mode **network** commands, connected routes are explicitly configured for advertisement into the BGP domain. The **network routemap** command provides the flexibility to change many BGP attributes. Refer to the *Border Gateway Protocol Configuration Mode Commands* section of the *Command Line Interface Reference* for details on these commands.

 **Important:** If a BGP task restarts because of a processing card failure, a migration, a crash, or the removal of a processing card, all peering session and route information is lost.

Configuring BGP

This section describes how to configure and enable basic BGP routing support in the system.

```
config
    context <context_name>
        router bgp <AS_number>
            neighbor <ip_address> remote-as <AS_num>
```

Notes:

- A maximum of 64 BGP peers are supported per context.
- Save your configuration as described in *Verifying and Saving Your Configuration*.

Redistributing Routes Into BGP (Optional)

Redistributing routes into BGP simply means that any routes from another protocol that meet a specified criterion, such as a route type, or a rule within a route-map, are redistributed through the BGP protocol to all BGP areas. This is an optional configuration.

```
config
    context <context_name>
        router bgp <as_number>
            redistribute bgp { bgp | connected | static } [ metric <metric_value> ]
            [ metric-type { 1 | 2 } ] [ route-map <route_map_name> ]
```

Notes:

- The redistribution options are connected, ospf, rip, or static. Refer to the *Border Gateway Protocol Configuration Mode Commands* section of the *Command Line Interface Reference* for details on the **redistribute** command.
- A maximum of 64 route-maps are supported per context.
- Save your configuration as described in *Verifying and Saving Your Configuration*.

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a network protocol used to detect faults between two forwarding engines connected by a link. BFD establishes a session between two endpoints over a particular link. If more than one link exists between two systems, multiple BFD sessions may be established to monitor each one of them. The session is established with a three-way handshake, and is torn down the same way. Authentication may be enabled on the session. A choice of simple password, MD5 or SHA1 authentication is available.

Overview of BFD Support

BFD does not have a discovery mechanism; sessions must be explicitly configured between endpoints. BFD may be used on many different underlying transport mechanisms and layers, and operates independently of all of these. Therefore, it needs to be encapsulated by whatever transport it uses.

Protocols that support some form of adjacency setup, such as OSPF or IS-IS, may also be used to bootstrap a BFD session. These protocols may then use BFD to receive faster notification of failing links than would normally be possible using the protocol's own keepalive mechanism.

In asynchronous mode, both endpoints periodically send Hello packets to each other. If a number of those packets are not received, the session is considered down.

When Echo is active, a stream of Echo packets is sent to the other endpoint which then forwards these back to the sender. Echo can be globally enabled via the **bfd-protocol** command, and/or individually enabled/disabled per interface. This function is used to test the forwarding path on the remote system.

The ASR 5000ASR 5000 supports BFD in asynchronous mode with optional Echo capability via static or BGP routing.

Configuring BFD

This section describes how to configure and enable basic BFD routing protocol support in the system.

There are several factors affecting the configuration of BFD protocol:

- [Configuring a BFD Context](#)
- [Configuring IPv4 BFD for Static Routes](#)
- [Configure IPv6 BFD for Static Routes](#)
- [Configuring BFD for Single Hop](#)
- [Configuring Multihop BFD](#)
- [Scaling of BFD](#)

Configuring a BFD Context

```
config
    context <context_name>
        bfd-protocol
            [ [no] bfd echo ]
    exit
```

Notes:

- Echo function can be optionally enabled for all interfaces in this context.
- 16 BFD sessions per context and 64 per chassis.

Configuring IPv4 BFD for Static Routes

Enable BFD on an interface.

```
config
    context <bfd_context_name>
        interface <if_name>
            ip address <ipv4_address> <ipv4_mask>
            bfd interval <interval_value> min_rx <rx_value> multiplier
            <multiplier_value>
            [ [no] bfd echo ]
        exit
```

Configure BFD static route.

```
ip route static bfd <if_name> <ipv4_gw_address>
```

Add static routes.

```
ip route <ipv4_address> <ipv4_mask>
```

```
ip route <ipv4_address> <ipv4_mask>
```

Configure IPv6 BFD for Static Routes

Enable BFD on an Interface

```
config
  context <bfd_context_name>
  interface <if_name>
    ipv6 address <ipv6_address> <ipv6_mask>
    bfd interval <interval_value> min_rx <rx_value> multiplier
    <multiplier_value>
    [ [no] bfd echo ]
  exit
```

Configure BFD static route.

```
ipv6 route static bfd <if_name> <ipv6_gw_address>
```

Add static routes.

```
ipv6 route <ipv6_address> <ipv6_mask>
ipv6 route <ipv6_address> <ipv6_mask>
```

Configuring BFD for Single Hop

Enable BFD on an interface.

```
config
  context <bfd_context_name>
  interface <if_name>
    ip address <ipv4_address> <ipv4_mask>
    ipv6 address <ipv6_address> <ipv6_mask>
    bfd interval <interval_value> min_rx <rx_value> multiplier
    <multiplier_value>
    [ [no] bfd echo ]
  exit
```

Enable BFD on a BGP Neighbor. See [Associate BGP Neighbors with the Context](#) for additional information.

Enable BFD on an OSPF Neighbor. See [Associate OSPF Neighbors with the Context](#) for additional information.

Configuring Multihop BFD

Enable BFD on an interface.

```

config

  context <bfd_context_name>

    interface <if_name>

      ip address <ipv4_address> <ipv4_mask>

      ipv6 address <ipv6_address> <ipv6_mask>

      bfd interval <interval_value> min_rx <rx_value> multiplier
      <multiplier_value>

      [ [no] bfd echo ]

    exit

```

Configure a Multihop BFD session.

```

bfd-protocol

  bfd multihop peer <destination-address> interval <interval-value>
multiplier <multiplier-value>

```

Enable BFD on a BGP Neighbor. See [Associate BGP Neighbors with the Context](#) for additional information.

Scaling of BFD

Configure an active BFD session using one of the above methods and use same BFD neighbor while configuring the active interface. See [Associate BFD Neighbor Groups with the BFD Protocol](#) for additional information.

```

bfd-protocol

  bfd nbr-group-name <grp_name> active-if-name <if_name> <nexthop_address>

```

Apply the same BFD results to one or more passive interfaces.

```

  bfd nbr-group-name <grp_name> passive-if-name <if_name>
  <nexthop_address>

  bfd nbr-group-name <grp_name> passive-if-name <if_name>
  <nexthop_address>

```

Associate BGP Neighbors with the Context

```
config
  context <context_name>
    router bgp <AS_number>
      neighbor <neighbor_ip-address> remote-as <rem_AS_number>
      neighbor <neighbor_ip-address> ebgp-multihop max-hop <max_hops>
      neighbor <neighbor_ip-address> update-source <update-src_ip-address>
      neighbor <neighbor_ip-address> failover bfd [ multihop ]
```

Notes:

- Repeat the sequence to add neighbors.

Associate OSPF Neighbors with the Context

```
config
  context <context_name>
    router ospf
      neighbor <neighbor_ip-address>
```

Notes:

- Repeat the sequence to add neighbors.

Associate BFD Neighbor Groups with the BFD Protocol

```
config
  context <context_name>
    bfd-protocol
      bfd nbr-group-name <grp_name> active-if-name <if_name> <nexthop_address>
      bfd nbr-group-name <grp_name> passive-if-name <if_name>
      <nexthop_address>
```



Important: Save your configuration as described in *Verifying and Saving Your Configuration*.

Appendix H

Session Recovery

With robust hardware failover and redundancy protection, any card-level hardware failures on the system can quickly be corrected. However, software failures can occur for numerous reasons, often without prior indication.

This segment describes the Session Recovery feature that provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault.



Important: Session Recovery is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

This segment includes the following sections:

- [How Session Recovery Works](#)
- [Additional Hardware Requirements](#)
- [Configuring the System to Support Session Recovery](#)

How Session Recovery Works

This section provides an overview of how this feature is implemented and the recovery process.

The Session Recovery feature provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system preventing a fully connected user session from being disconnected.

Session recovery is performed by mirroring key software processes (for example, session manager and AAA manager) within the system. These mirrored processes remain in an idle state (standby-mode) wherein they perform no processing, until they may be needed in the event of a software failure (for example, a session manager task aborts).

The system spawns new instances of “standby mode” session and AAA managers for each active control processor (CP) being used. These mirrored processes require both memory and processing resources, which means that additional hardware may be required to enable this feature (see the [Additional Hardware Requirements](#) section).

Other key system-level software tasks, such as VPN manager, are performed on a physically separate packet processing card to ensure that a double software fault (for example, session manager and VPN manager fails at same time on same card) cannot occur. The packet processing card that hosts the VPN manager process is in active mode and reserved by the operating system for this sole use when session recovery is enabled.

There are two modes of session recovery.

- **Task recovery mode:** Wherein one or more session manager failures occur and are recovered without the need to use resources on a standby packet processing card. In this mode, recovery is performed by using the mirrored “standby-mode” session manager task(s) running on active packet processing cards. The “standby-mode” task is renamed, made active, and is then populated using information from other tasks such as AAA manager. In case of Task failure, limited subscribers will be affected and will suffer outage only until the task starts back up.
- **Full packet processing card recovery mode:** Used when a packet processing card hardware failure occurs, or when a planned packet processing card migration fails. In this mode, the standby packet processing card is made active and the “standby-mode” session manager and AAA manager tasks on the newly activated packet processing card perform session recovery.

Session/Call state information is saved in the peer AAA manager task because each AAA manager and session manager task is paired together. These pairs are started on physically different packet processing cards to ensure task recovery.

There are some situations wherein session recovery may not operate properly. These include:

- Additional software or hardware failures occur during the session recovery operation. For example, an AAA manager fails while the state information it contained was being used to populate the newly activated session manager task.
- A lack of hardware resources (packet processing card memory and control processors) to support session recovery.

 **Important:** After a session recovery operation, some statistics, such as those collected and maintained on a per manager basis (AAA Manager, Session Manager, etc.) are in general not recovered, only accounting and billing related information is checkpointed and recovered.

Session Recovery is available for the following functions:

- Any session needing L2TP LAC support (excluding regenerated PPP on top of an HA or GGSN session)
- ASR 5000 only – Closed RP PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP

- CSCF sessions
- GGSN services for IPv4 and PPP PDP contexts
- HA services supporting Mobile IP and/or Proxy Mobile IP session types with or without per-user Layer 3 tunnels
- ASR 5000 only – **HNB-GW**: HNB Session over IuH
- ASR 5000 only – **HNB-GW**: HNB-CN Session over IuPS and IuCS
- ASR 5000 only – **HNB-GW**: SeGW Session IPSec Tunnel
- ASR 5000 only – HSGW services for IPv4
- ASR 5000 only – IPSG-only systems
- LNS session types
- PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- P-GW services for IPv4
- ASR 5000 only – SGSN services (3G and 2.5G services) for IPv4 and PPP PDP contexts

Session recovery is **not supported** for the following functions:

- Destination-based accounting recovery
- GGSN network initiated connections
- GGSN session using more than 1 service instance
- MIP/L2TP with IPSec integration
- MIP session with multiple concurrent bindings
- Mobile IP sessions with L2TP
- Multiple MIP sessions

 **Important:** Always refer to the Administration Guides for individual products for other possible session recovery and Interchassis Session Recovery (ICSR) support limitations.

When session recovery occurs, the system reconstructs the following subscriber information:

- Data and control state information required to maintain correct call behavior.
- A minimal set of subscriber data statistics; required to ensure that accounting information is maintained.
- A best-effort attempt to recover various timer values such as call duration, absolute time, and others.
- The idle time timer is reset to zero and the re-registration timer is reset to its maximum value for HA sessions to provide a more conservative approach to session recovery.

 **Important:** Any partially connected calls (for example, a session where HA authentication was pending but has not yet been acknowledged by the AAA server) are not recovered when a failure occurs.

Additional Hardware Requirements

Because session recovery requires numerous hardware resources, such as memory, control processors, NPU processing capacity, some additional hardware may be required to ensure that enough resources are available to fully support this feature.



Important: A minimum of four packet processing cards (three active and one standby) per individual chassis is required to use this feature.

To allow for complete session recovery in the event of a hardware failure during a packet processing card migration, a minimum of three active packet processing cards and two standby packet processing cards should be deployed.

To assist you in your network design and capacity planning, consider the following factors:

- Subscriber capacity is decreased depending on the hardware configuration. A fully configured chassis would experience a smaller decrease in subscriber capacity versus a minimally configured chassis.
- The amount by which control transaction processing capacity is reduced.
- The reduction in subscriber data throughput.
- The recovery time for a failed software task.
- The recovery time for a failed packet processing card.

A packet processing card migration may temporarily impact session recovery as hardware resources (memory, processors, etc.) that may be needed are not available during the migration. To avoid this condition, a minimum of two standby packet processing cards should be configured.

Configuring the System to Support Session Recovery

The following procedures allow you to configure the session recovery feature for either an operational system that is currently in-service (able to accept incoming calls) or a system that is out-of-service (not part of your production network and, therefore, not processing any live subscriber/customer data).

 **Important:** The session recovery feature, even when the feature use key is present, is disabled by default on the system.

Enabling Session Recovery

As noted earlier, session recovery can be enabled on a system that is out-of-service (OOS) and does not yet have any contexts configured, or on an in-service system that is currently capable of processing calls. However, if the system is in-service, it must be restarted before the session recovery feature takes effect.

Enabling Session Recovery on an Out-of-Service System

The following procedure is for a system that does not have any contexts configured.

To enable the session recovery feature on an out-of-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

Step 1 At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command.

 **Important:** If the current status of the Session Recovery feature is Disabled, you cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```
configure
    require session recovery
end
```

Step 3 Save your configuration as described in *Verifying and Saving Your Configuration*.

The system, when started, enables session recovery, creates all mirrored “standby-mode” tasks, and performs packet processing card reservations and other operations automatically.

Step 4 After the system has been configured and placed in-service, you should verify the preparedness of the system to support this feature as described in [Viewing Session Recovery Status](#).

Enabling Session Recovery on an In-Service System

When enabling session recovery on a system that already has a saved configuration, the session recovery commands are automatically placed before any service configuration commands in the configuration file.

To enable the session recovery feature on an in-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

- Step 1** At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command:

 **Important:** If the current status of the Session Recovery feature is Disabled, You cannot enable this feature until a license key is installed in the system.

- Step 2** Use the following configuration example to enable session recovery.

```
configure
  require session recovery
end
```

 **Important:** This feature does not take effect until after the system has been restarted.

- Step 3** Save your configuration as described in *Verifying and Saving Your Configuration*.

- Step 4** Perform a system restart by entering the **reload** command:

The following prompt appears:

```
Are you sure? [Yes|No]:
```

Confirm your desire to perform a system restart by entering **yes**.

The system, when restarted, enables session recovery and creates all mirrored “standby-mode” tasks, performs packet processing card reservations, and other operations automatically.

- Step 5** After the system has been restarted, you should verify the preparedness of the system to support this feature as described in [Viewing Session Recovery Status](#).

 **Important:** More advanced users may opt to simply insert the **require session recovery** command syntax into an existing configuration file using a text editor or other means, and then applying the configuration file manually. Exercise caution when doing this to ensure that this command is placed among the first few lines of any existing configuration file; it must appear before the creation of any non-local context.

Disabling the Session Recovery Feature

To disable the session recovery feature on a system, enter the **no require session recovery** command from the Global Configuration mode prompt.

 **Important:** If this command is issued on an in-service system, then the system must be restarted by issuing the **reload** command.

Viewing Session Recovery Status

To determine if the system is capable of performing session recovery, when enabled, enter the **show session recovery status verbose** command from the Exec mode prompt.

The output of this command should be similar to the examples shown below.

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status      : SESSMGR Not Ready For Recovery
  Last Status Update  : 1 second ago
```

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status      : Ready For Recovery
  Last Status Update  : 8 seconds ago
```

```
[local]host_name# show session recovery status verbose
Session Recovery Status:
  Overall Status      : Ready For Recovery
  Last Status Update  : 2 seconds ago
```

cpu state	----sessmgr---		----aaamgr----		demux		status
	active	standby	active	standby	active		
1/1 Active	2	1	1	1	0		Good
1/2 Active	1	1	0	0	0		Good
1/3 Active	1	1	3	1	0		Good
2/1 Active	1	1	1	1	0		Good
2/2 Active	1	1	0	0	0		Good
2/3 Active	2	1	3	1	0		Good
3/0 Active	0	0	0	0	1		Good (Demux)
3/2 Active	0	0	0	0	1		Good (Demux)
4/1 Standby	0	2	0	1	0		Good
4/2 Standby	0	1	0	0	0		Good
4/3 Standby	0	2	0	3	0		Good

```
[local]host_name#
```

Viewing Recovered Session Information

To view session state information and any session recovery status, enter the following command:

```
show subscriber debug-info { callid id | msid id | username name }
```

Keyword/Variable	Description
callid <i>id</i>	Displays subscriber information for the call specified by <i>id</i> . The call ID is an 8-byte hexadecimal number.
msid <i>id</i>	Displays information for the mobile user identified by an MSID from 7 to 16 digits specified as an IMSI, MIN, or RMI. Wildcard characters \$ and * are allowed. The * wildcard matches multiple characters and the \$ wildcard matches a single character. If you do not want the wildcard characters interpreted as a wildcard enclose them in single quotes ('). For example; '\$'.
username <i>name</i>	Displays information for connections for the subscriber identified by a previously configured username. <i>name</i> is a sequence of alphanumeric characters and/or wildcard characters ('\$ and '*') from 1 to 127 characters in length. The * wildcard matches multiple characters and the \$ wildcard matches a single character. If you do not want the wildcard characters interpreted as wildcard enclose them in single quotes ('). For example; '\$'.

The following example shows the output of this command both before and after a session recovery operation has been performed. The “Redundancy Status” fields in this example have been bold-faced for clarity.

```
username: user1      callid: 01callb1      msid: 0000100003
Card/Cpu: 4/2
Sessmgr Instance: 7
Primary callline:
Redundancy Status: Original Session
Checkpoints   Attempts   Success   Last-Attempt   Last-Success
  Full:             69         68         29800ms        29800ms
  Micro:            206        206         20100ms        20100ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
      State                               Event
SMGR_STATE_OPEN                          SMGR_EVT_NEWCALL
SMGR_STATE_NEWCALL_ARRIVED                SMGR_EVT_ANSWER_CALL
SMGR_STATE_NEWCALL_ANSWERED               SMGR_EVT_LINE_CONNECTED
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_LINK_CONTROL_UP
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_AUTH_REQ
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_IPADDR_ALLOC_SUCCESS
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_UPDATE_SESS_CONFIG
SMGR_STATE_LINE_CONNECTED                  SMGR_EVT_LOWER_LAYER_UP

Data Reorder statistics
Total timer expiry:                        0          Total flush (tmr expiry):      0
  Total no buffers:                        0          Total flush (no buffers):      0
  Total flush (queue full):                0          Total flush (out of range):    0
  Total flush (svc change):                0          Total out-of-seq pkt drop:    0
  Total out-of-seq arrived:                0

IPv4 Reassembly Statistics:
  Success:                                0          In Progress:                   0
  Failure (timeout):                       0          Failure (no buffers):          0
  Failure (other reasons):                 0
```

```

Redirected Session Entries:
  Allowed:                2000      Current:                0
  Added:                  0         Deleted:                0
  Revoked for use by different subscriber: 0
Peer callline:
Redundancy Status: Original Session
Checkpoints   Attempts   Success   Last-Attempt   Last-Success
  Full:        0         0         0ms            0ms
  Micro:       0         0         0ms            0ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State                               Event
  SMGR_STATE_LINE_CONNECTED          SMGR_EVT_LOWER_LAYER_UP
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_REQ
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_SUCCESS
  SMGR_STATE_CONNECTED               SMGR_EVT_REQ_SUB_SESSION
  SMGR_STATE_CONNECTED               SMGR_EVT_RSP_SUB_SESSION
  SMGR_STATE_CONNECTED               SMGR_EVT_ADD_SUB_SESSION
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_REQ
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_SUCCESS
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_REQ
  SMGR_STATE_CONNECTED               SMGR_EVT_AUTH_SUCCESS
Data Reorder statistics
  Total timer expiry:                0         Total flush (tmr expiry): 0
  Total no buffers:                   0         Total flush (no buffers): 0
  Total flush (queue full):           0         Total flush (out of range):0
  Total flush (svc change):           0         Total out-of-seq pkt drop: 0
  Total out-of-seq arrived:           0
IPv4 Reassembly Statistics:
  Success:                            0         In Progress:            0
  Failure (timeout):                   0         Failure (no buffers):   0
  Failure (other reasons):             0
Redirected Session Entries:
  Allowed:                2000      Current:                0
  Added:                  0         Deleted:                0
  Revoked for use by different subscriber: 0

```

Notice that in the example above, where the session has been recovered/recreated, that state events (FSM Event State field) no longer exist. This field is re-populated as new state changes occur.

Appendix I

VLANs

This segment provides information on configuring virtual local area networks (VLANs) in support of enhanced or extended services. The product administration guides provide examples and procedures for configuration of services on the system that may utilize VLANs. You should select the configuration example that best meets your service model before using the procedures described below.

This segment includes the following sections:

- [Overview](#)
- [Creating VLAN Tags](#)
- [Configuring Subscriber VLAN Associations](#)
- [Verify the Port Configuration](#)
- [VLAN-Related CLI Commands](#)

 **Important:** VLAN – Layer 2 Traffic Management is a Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

Overview

Virtual LANs (VLANs) provide greater flexibility in the configuration and use of contexts and services.

They are configured as “tags” on a per-port basis and allow more complex configurations to be implemented. The VLAN tag allows a single physical port to be bound to multiple logical interfaces that can be configured in different contexts. Therefore, each Ethernet port can be viewed as containing many logical ports when VLAN tags are employed.



Important: VLANs are supported in conjunction with subscriber traffic ports on Management I/O (MIO/UMIO) cards. The system supports the configuration limits for VLANs as described in *Engineering Rules*.

Overlapping IP Address Pool Support – GGSN

Overlapping IP Address pools provides allow operators to more flexibly support multiple corporate VPN customers with the same private IP address space without expensive investments in physically separate routers or virtual routers.

The system supports two types of overlapping pools – resource and overlap. Resource pools are designed for dynamic assignment only, and use a VPN tunnel (such as a GRE tunnel) to forward and receive the private IP addresses to and from the VPN. Overlapping type pools can be used for both dynamic and static addressing, and use VLANs and a next hop forwarding address to connect to the VPN customer

To forward downstream traffic to the correct PDP context, the GGSN uses either the GRE tunnel ID or the VLAN ID to match the packet. When forwarding traffic upstream, the GGSN uses the tunnel and forwarding information in the IP pool configuration; overlapping pools must be configured in the APN in such instances.

When a PDP context is created, the IP address is assigned from the IP pool. In this case the forwarding rules are also configured into the GGSN. If the address is assigned statically, when the GGSN confirms the IP address from the pool configured in the APN, the forwarding rules are also applied.

The GGSN can scale to as many actual overlapping pools as there are VLAN interfaces per context, and there can be multiple contexts per GGSN. The limit is the number of IP pools. This scalability allows operators who wish to provide VPN services to customers using the customer's private IP address space, not to be concerned about escalating hardware costs or complex configurations.

RADIUS VLAN Support – Enhanced Charging Services

VPN customers often use private address space which can easily overlap with other customers. The subscriber addresses are supported with overlapping pools which can be configured in the same virtual routing context.

RADIUS Server and NAS IP addresses do not need to be in separate contexts, thereby simplifying APN and RADIUS configuration and network design. This feature allows the following scenarios to be defined in the same context:

- Overlapping RADIUS NAS-IP addresses for various RADIUS server groups representing different APNs.
- Overlapping RADIUS server IP addresses for various RADIUS servers groups.

Every overlapping NAS-IP address is given a unique next-hop address which is then bound to an interface that is bound to a unique VLAN, thereby allowing the configuration to exist within the same context.

The system forwards RADIUS access requests and accounting messages to the next hop defined for that NAS-IP; the connected routers forward the messages to the RADIUS server. The next hop address determines the interface and VLAN to use. Traffic from the server is identified as belonging to a certain NAS-IP by the port/VLAN combination.

The number of RADIUS NAS-IP addresses that can be configured is limited by the number of loopback addresses that can be configured.

APN Support – PDN Gateway (P-GW)

P-GW Access Point Name (APN) supports extensive parameter configuration flexibility for the APN. VLAN tagging may be selected by the APN, but are configured in the P-GW independently from the APN.

Creating VLAN Tags

Use the following example to create VLANs on a port and bind them to pre-existing interfaces. For information on creating interfaces, refer to *System Interfaces and Ports*.

```
config
  port ethernet <slot/port>
    no shutdown
    vlan <vlan_tag_ID>
    no shutdown
    bind interface <interface_name> <context_name>
  end
```

Notes:

- *Optional:* Configure VLAN-subscriber associations. Refer to [Configuring Subscriber VLAN Associations](#) for more information.
- Repeat this procedure as needed to configure additional VLANs for the port.
- Refer to [VLAN-Related CLI Commands](#) and the *Command Line Interface Reference* for additional information.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying the Port Configuration

Run the following command to verify the port configuration:

```
show port info slot/port
```

An example of this command's output when at least one VLAN has been configured for the port is shown below:

```
Port: 5/11
  Port Type           : 10Gig Ethernet
  Role                : Service Port
  Description         : (None Set)
  Redundancy Mode     : Port Mode
  Redundant With      : 6/11
  Preferred Port      : Non-Revertive
  Physical ifIndex    : 85262336
  Administrative State : Enabled
  Configured Duplex   : Auto
  Configured Speed    : Auto
  Fault Unidirection Mode : 802_3ae clause 46
  Configured Flow Control : Enabled
  Interface MAC Address : 02-05-47-C1-32-DA
  Fixed MAC Address    : 02-05-47-C1-32-BA
  Boxer Interface TAP : None
  Link State          : Up
  Link Duplex         : Unknown
  Link Speed          : Unknown
  Flow Control        : Disabled
  Link Aggregation Group : None
  Untagged:
    Logical ifIndex   : 85262337
    Operational State : Up, Active
  Tagged VLAN: VID 10
    Logical ifIndex   : 285278210
    VLAN Type         : Subscriber
    Administrative State : Enabled
    Operational State : Up, Active
  Number of VLANs    : 1
  SFP Module         : 10G Base-SR
```

Notes:

- Repeat this sequence as needed to verify additional ports.
- *Optional:* Configure VLAN-subscriber associations. Refer to [Configuring Subscriber VLAN Associations](#) for more information.
- Refer to [VLAN-Related CLI Commands](#) for additional information.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Subscriber VLAN Associations

Subscriber traffic can be routed to specific VLANs based on the configuration of their user profile. This functionality provides a mechanism for routing all traffic from a subscriber over the specified VLAN. All packets destined for the subscriber must also be sent using only IP addresses valid on the VLAN or they will be dropped.

RADIUS Attributes Used

The following RADIUS attributes can be configured within subscriber profiles on the RADIUS server to allow the association of a specific VLAN to the subscriber:

- **SN-Assigned-VLAN-ID:** In the Starent VSA dictionary
- **SN1-Assigned-VLAN-ID:** In the Starent VSA1 dictionary

 **Important:** Since the instructions for configuring subscriber profiles differ between RADIUS server applications, this section only describes the individual attributes that can be added to the subscriber profile. Please refer to the documentation that shipped with your RADIUS server for instructions on configuring subscribers.

Configuring Local Subscriber Profiles

Use the configuration example below to configure VLAN associations within local subscriber profiles on the system.

 **Important:** These instructions assume that you have already configured subscriber-type VLAN tags according to the instructions provided in [Creating VLAN Tags](#).

```
config
    context <context_name>
        subscriber name <user_name>
            ip vlan <vlan_id>
        end
```

Verify the Subscriber Profile Configuration

Use the following command to view the configuration for a subscriber profile:

```
show subscriber configuration username user_name
```

Notes:

- Repeat this command for each subscriber.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

VLAN-Related CLI Commands

VLAN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VLAN-related functions.

For detailed information regarding the use of the commands listed below, see the *Command Line Interface Reference*.

Table 53. VLAN-Related Configuration Commands

CLI Mode	Command	Description
AAA Server Group Configuration Mode	radius attribute nas-ip-address address <ip_address> nexthop-forwarding-address <ip_address> vlan <vlan_id>	Sets the RADIUS client to provide the VLAN ID with the nexthop forwarding address to a system when running in single nexthop gateway mode. Note: To access the vlan keyword, aaa-large configuration must be enabled via the Global Configuration mode.
ACS Charging Action Configuration Mode	ip vlan <vlan_id>	Configures the VLAN identifier to be associated with the subscriber traffic in the destination context.
Context Configuration Mode	ip pool <pool_name> nexthop forwarding address <ip_address> overlap vlanid <vlan_id>	When a nexthop forwarding address is configured, the overlap vlanid keyword enables support for overlapping IP address pools and associates the pool with the specified VLAN ID.
Context Configuration Mode	ip routing overlap-pool	Advertises overlap-pool addresses in dynamic routing protocols when overlap pools are configured using VLAN IDs. When enabled, the overlap addresses are added as interface addresses and advertised.
Context Configuration Mode	radius attribute nas-ip-address address <ip_address> nexthop-forwarding-address <ip_address> vlan <vlan_id>	Specifies the VLAN ID to be associated with the next-hop IP address.
Ethernet Interface Configuration Mode	[no] logical-port-statistics	Enables or disables the collection of logical port (VLAN and NPU) bulk statistics for the first 32 configured Ethernet or PVC interface types.
Ethernet Interface Configuration Mode	vlan-map next-hop <ipv4_address>	Sets a single next-hop IP address so that multiple VLANs can use a single next-hop gateway. The vlan-map is associated with a specific interface.
Ethernet Port Configuration Mode	vlan <vlan_id>	Enters VLAN Configuration mode.
PVC Configuration Mode	[no] shutdown	Enables or disables traffic over a specified VLAN. See below.
Subscriber Configuration Mode	ip vlan <vlan_id>	Configures the subscriber VLAN ID that is used with the assigned address for the subscriber session to receive packets. If the IP pool from which the address is assigned is configured with a VLAN ID, this subscriber configured VLAN ID overrides it.

CLI Mode	Command	Description
VLAN Configuration Mode	bind interface <interface_name> <context_name>	Binds a virtual interface and context to support VLAN service.
VLAN Configuration Mode	[no] ingress-mode	Enables or disables port ingress (incoming) mode.
VLAN Configuration Mode	priority <value>	Configures an 802.1p VLAN priority bit for ASN-GW service only.
VLAN Configuration Mode	[no] shutdown	Enables or disables traffic over the current VLAN.
VLAN Configuration Mode	vlan-map interface <if_name> <context_name>	Associates an IP interface having a VLAN ID with a context.

Table 54. VLAN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show commands	clear port <slot/port> vlan <vlan_id>	Clears NPU statistics for the port that has a previously configured VLAN ID.
Exec Mode show commands	show logical-port utilization table vlan { 5-minute hourly }	Displays VLAN utilization for a specified collection interval.
Exec Mode show commands	show port info <slot/port> vlan <vlan_id>	Displays NPU counters for a previously configured VLAN ID.

Appendix J

BGP MPLS VPNs

This segment describes services that are supported by the ASR 5x00 for Border Gateway Protocol (BGP) Multi-Protocol Label Switching (MPLS) Virtual Private Networks (VPNs).

 **Important:** MPLS is a licensed Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

It includes the following topics:

- [Introduction](#)
- [MPLS-CE Connected to PE](#)
- [ASR 5x00 as a PE](#)
- [VPN-Related CLI Commands](#)

Introduction

Service providers require the ability to support a large number of corporate Access Point Names (APNs) which have a number of different addressing models and requirements. The ASR 5x00 uses BGP MPLS Layer 3 VPNs to segregate corporate customer APNs in a highly scalable manner. This solution conforms to RFC 4364 – *BGP/MPLS IP Virtual Private Networks (VPNs)*.

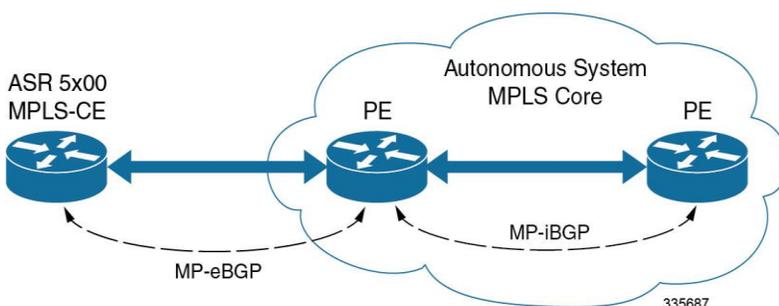
The BGP/MPLS solution supports the following scenarios:

- [MPLS-CE Connected to PE](#)
- [ASR 5x00 as a PE](#)

MPLS-CE Connected to PE

In this scenario the ASR 5x00 functions as an MPLS-CE (Customer Edge) network element connected to a Provider Edge (PE) Label Edge Router (LER), which in turn connects to the MPLS core (RFC 4364). See the figure below.

Figure 22. ASR 5x00 MPLS-CE to PE



The MPLS-CE functions like a PE router within its own Autonomous System (AS). It maintains Virtual Routing and Forwarding (VRF) routes and exchanges VPN route information with the PE via an MP-eBGP (Multi-Protocol-external BGP) session.

The PE is also configured with VRFs and exchanges VPN routes with other PEs in its AS via MP-iBGP (Multi-Protocol-internal BGP) connections and the MPLS-CE via an MP-eBGP connection.

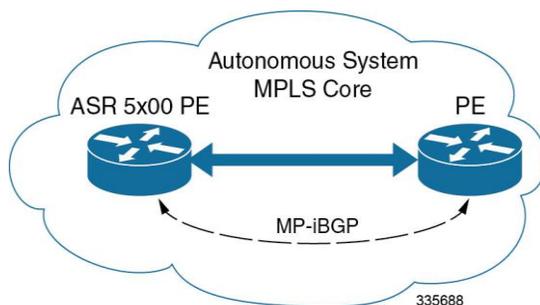
The EBGP connection allows the PE to change next-hop IP addresses and labels in the routes learned from IBGP peers before advertising them to the MPLS-CE. The MPLS-CE in this case uses only MP-eBGP to advertise and learn routes. Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) are not required because of direct-connect EBGP peering. The MPLS-CE in this scenario pushes/pops a single label (learned over the MP-eBGP connection) to/from the PE.

ASR 5x00 as a PE

Overview

In this scenario, the ASR 5x00 functions as a PE router sitting at the edge of the MPLS core. See the figure below.

Figure 23. ASR 5x00 as a PE



The ASR 5x00 eliminates the need for an ASBR or PE as shown in the first two scenarios. In this scenario, two main requirements are introduced: IBGP functionality and MPLS label distribution protocols.

The ASR 5x00 can be configured to add two labels:

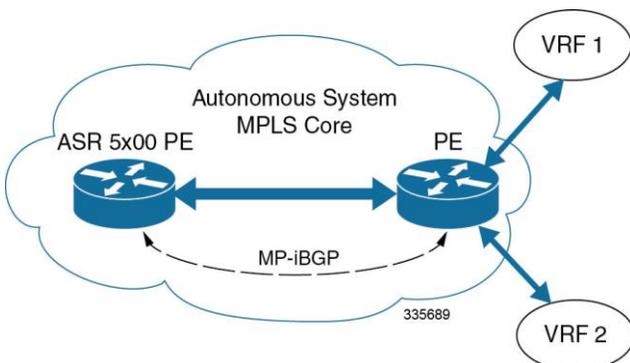
- an outer label learned from LDP or RSVP-TE (RSVP-Traffic Engineering)
- an inner label learned from MP-iBGP

This solution supports traffic engineering and QoS initiated via the ASR 5x00.

Sample Configuration

In this example, VRFs are configured on the ASR 5x00 PE and pools are associated with VRFs. The ASR 5x00 exchanges VPN routes with its IBGP peers (PE routers) and learns the MPLS paths to reach PEs via LDP. The ASR 5x00 forwards the packets to the next-hop with two labels – an inner label learned from PE and an outer label learned from the next hop IBGP neighbor.

Figure 24. Sample Configuration



```

mpls ip
  protocol ldp
  enable
  exit
exit

ip vrf vrf1
  mpls traffic-class copy
exit
ip vrf vrf2
  mpls traffic-class value 5
exit

router bgp 300
  ip vrf vrf1
    route-target export 300 1
    route-target import 300 1
    route-distinguisher 300 1
  exit
  ip vrf vrf2
    route-target export 300 2
    route-target import 300 2
    route-distinguisher 300 2
  exit

router-id 2.2.2.2
neighbor 192.168.107.20 remote-as 300
neighbor 192.168.107.20 update-source node1_loopback

address-family vpnv4
  neighbor 192.168.107.20 activate
  neighbor 192.168.107.20 send-community both
  neighbor 192.168.107.20 next-hop-self
exit

address-family ipv4 vrf vrf1
  redistribute connected

```

```
exit

address-family ipv4 vrf vrf2
  redistribute connected
exit

interface interface_to_internet
  ip address 192.168.109.65/24
  mpls ip
exit
router ospf
  network 192.168.109.0/24 area 0.0.0.0
exit
```

VPN-Related CLI Commands

VPN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VPN-related functions.

For detailed information regarding the use of the commands listed below, see the *Command Line Interface Reference*.

Table 55. VPN-Related Configuration Commands

CLI Mode	Command	Description
BGP Address-Family (IPv4/IPv6) Configuration Mode	neighbor <ip_address> activate	Enables the exchange of routing information with a peer router.
BGP Address-Family (IPv4/IPv6) Configuration Mode	neighbor <ip_address> send community { both extended standard }	Sends the community attributes to a peer router (neighbor).
BGP Address-Family (IPv4/IPv6) Configuration Mode	redistribute connected	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Address-Family (VPNv4) Configuration Mode	neighbor <ip_address> activate	Enables the exchange of routing information with a peer router.
BGP Address-Family (VPNv4) Configuration Mode	neighbor <ip_address> send community { both extended standard }	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	neighbor <ip_address> activate	Enables the exchange of routing information with a peer router.
BGP Address-Family (VRF) Configuration Mode	neighbor <ip_address> send community { both extended standard }	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	redistribute connected	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Configuration Mode	address-family { ipv4 vrf <vrf_name> vpn4 }	Enables the exchange of IPv4 VRF routing information. There is a different mode for each address-family.
BGP Configuration Mode	ip vrf <vrf_name>	Adds a VRF to BGP and switches to the VRF Configuration mode to allow configuration of BGP attributes for the VRF.
BGP IP VRF Configuration Mode	route-distinguisher { <as_value> <ip_address> } <rd_value>	Assigns a Route Distinguisher (RD) for the VRF. The RD value must be a unique value on the router for each VRF.

CLI Mode	Command	Description
BGP IP VRF Configuration Mode	route-target { both import export } { <as_value> <ip_address> } <rt_value>	Adds a list of import and export route-target extended communities to the VRF.
Context Configuration Mode	ip pool <pool_name> <addr_range> vrf <vrf_name> [mpls-label input <inlabel1> output <outlabel1> <outlabel2>]	Configures a pool into the specified VRF. This parameter must be specified with the Next-Hop parameter. <i>inlabel1</i> is the MPLS label that identifies inbound traffic destined for this pool. <i>outlabel1</i> and <i>outlabel2</i> specify the MPLS labels to be added to packets sent for subscribers from this pool.
Context Configuration Mode	ip vrf <vrf_name>	Creates a VRF and assigns a VRF-ID. A VRF is created in the router.
Context Configuration Mode	mpls bgp forwarding	Globally enables MPLS Border Gateway Protocol (BGP) forwarding.
Context Configuration Mode	mpls ip	Globally enables the MPLS forwarding of IPv4 packets along normally routed paths.
Context Configuration Mode	radius change-authorize-nas-ip ip_address <ip_address> { encrypted key } <value> port <port_num> mpls input <inlabel> output <outlabel1> <outlabel2>	Configures COA traffic to use the specified MPLS labels. <i>inlabel</i> identifies inbound COA traffic. <i>outlabel1</i> and <i>outlabel2</i> specify the MPLS labels to be added to the COA response. <i>outlabel1</i> is the inner output label; <i>outlabel2</i> is the outer output label.
Ethernet Interface Configuration Mode	mpls ip	Enables dynamic MPLS forwarding of IP packets on this interface.
Exec Mode	clear ip bgp peer	Clears BGP sessions.
Exec Mode	lsp-ping <ip_prefix_FEC>	Checks MPLS Label-Switched Path (LSP) connectivity for the specified forwarding equivalence class (FEC). It must be followed by an IPv4 or IPv6 FEC prefix.
Exec Mode	lsp-traceroute <ip_prefix_FEC>	Discovers MPLS LSP routes that packets actually take when traveling to their destinations. It must be followed by an IPv4 or IPv6 FEC prefix.
IP VRF Context Configuration Mode	mpls map-dscp-to-exp dscp <dscp_bit_value> exp <exp_bit_value>	Maps the final differentiated services code point (DSCP) bit value in the IP packet header to the final Experimental (EXP) bit value in the MPLS header for incoming traffic.
IP VRF Context Configuration Mode	mpls map-exp-to-dscp exp <exp_bit_value> dscp <dscp_bit_value>	Maps the incoming EXP bit value in the MPLS header to the internal DSCP bit value in IP packet headers for outgoing traffic.
MPLS-IP Configuration Mode	protocol ldp	Creates the MPLS protocol family configuration modes, or configures an existing protocol and enters the MPLS-LDP Configuration Mode in the current context. This command configures the protocol parameters for the MPLS protocol family.

CLI Mode	Command	Description
MPLS-LDP Configuration Mode	advertise-labels { explicit-null implicit-null }	Configure advertisement of Implicit NULL or Explicit NULL label for all the prefixes advertised by the system in this context.
MPLS-LDP Configuration Mode	discovery { hello { hello-interval <seconds> hold-interval <seconds> } transport-address <ip_address> }	Configures the Label Distribution Protocol (LDP) neighbor discovery parameters.
MPLS-LDP Configuration Mode	enable	Enables Label Distribution Protocol (LDP).
MPLS-LDP Configuration Mode	router-id <ip_address>	Configures the LDP Router ID.
MPLS-LDP Configuration Mode	session timers { hold-interval <seconds> keepalive-interval <seconds> }	Configures the LDP session parameters.

Table 56. VPN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show Commands	show ip bgp neighbors	Displays information regarding BGP neighbors.
Exec Mode show Commands	show ip bgp vpnv4 { all route-distinguisher vrf }	Displays all VPNv4 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	show ip pool	Displays pool details including the configured VRF.
Exec Mode show Commands	show mpls cross-connect	Displays MPLS cross-connect information. MPLS tunnel cross-connects between interfaces and Label-Switched Paths (LSPs) connect two distant interface circuits of the same type via MPLS tunnels that use LSPs as the conduit.
Exec Mode show Commands	show mpls ftm [vrf <vrf_name>]	Displays MPLS FEC-to-NHLFE (FTN) table information.
Exec Mode show Commands	show mpls ftm [vrf <vrf_name>]	Displays contents of the MPLS FTN table for a specified VRF.
Exec Mode show Commands	show mpls ilm	Displays MPLS Incoming Label Map (ILM) table information.
Exec Mode show Commands	show mpls ldp	Displays the MPLS LDP information.
Exec Mode show Commands	show mpls nexthop-label-forwarding-entry	Displays MPLS Next-Hop Label Forwarding Entry (NHLFE) table information.