Avaya Secure Router 1000 Series Routing Guide

9.4
NN47262-101, 02.01
December 2010

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its Hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support Web site: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

Copyright

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://support.avaya.com/Copyright.

Trademarks

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation and Product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners, and “Linux” is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support Web site: http://support.avaya.com.

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your Product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site: http://support.avaya.com.

Contents

Chapter 1: ABOUT THIS GUIDE

This guide describes Avaya Secure Router 1000 implementation and command usage of BGP4, OSPF, RIP, and multicast routing protocols. It is designed for network managers and administrators who are responsible for the operation of LAN and WAN equipment and are familiar with networks and routing protocols.

Organization

The following table describes the organization and content of this guide.

Table 1: Guide Organization: Major Sections

About This Guide	Defines the user audience, describes the document's organization, introduces special notices, and provides information about other Avaya Secure Router user guides.
Protocols Overview	Provides high-level information about the routing protocols supported by Avaya.
Command Line Interface	Describes the command line interface (CLI) and how to access navigation and help features. A review of Secure Router configuration standards is included.
Commands	Describes the individual CLI commands. Commands are organized by protocol and are in alphabetical order. Each entry provides a command description and syntax and usage examples.

Documentation

Avaya user guides, which are provided in portable document format (PDF), are included on the Avaya Secure Router Documentation CD-ROM that ships with the Models 1001, 1001S, 1002, and 1004 router. The PDF files are also available on the Avaya website: http://www.avaya.com

To view PDF files, Adobe Acrobat® Reader® 4.0, or newer, must be installed on your workstation. If you do not have the Adobe Acrobat Reader installed on your system, you can obtain it free from the Adobe website: http://www.adobe.com.

About the Avaya Secure Router Documentation CD

This product ships with a CD that includes the following documentation:

- *Avaya Secure Router 1000 Quick Start Guide*
- *Avaya Secure Router 1000 Installation Guide*
- *Avaya Secure Router 1000 Command Reference Guide*
- *Avaya Secure Router 1000 Routing Guide*
- *Avaya Secure Router 1000 Configuration Guide*
- *Avaya Secure Router 1000 Web UI User Guide*
- Supported standard and enterprise MIBs
- Feature summaries
- SNMP trap descriptions with default configurations

Navigation

Upon inserting the Avaya Secure Router Documentation CD into your CD-ROM drive. Click a link to open a pdf version of the target document. If you do not have Adobe Acrobat (version 4.0, or later) or Acrobat Reader installed on your PC, click the Adobe button on the navigation screen to go to the Adobe website, where you can download a free copy of the Acrobat Reader application.

If a browser session is not opened, click "Start\Run," enter the drive letter of your CD-ROM drive in the "Open" entry box, and click "OK."

Printing Documents

To print any pdf document on the CD, follow this procedure.

1. Open the desired document by clicking the document link in the CD navigation window.
2. Click the "Printer" icon on the Adobe Acrobat tool bar.
3. In the "Windows Print" dialog box, select a local default printer in the "Printers" drop down selection box.
4. Click "OK."

   The following list includes other available and related documentation.

   - *Release Notes*

Printed release notes provide the latest information. If release notes are provided with your product, follow the instructions contained within them in addition to those provided in other documentation.

- *Avaya Secure Router 1000 Quick Start Guide*

  This guide is designed for advanced users who need minimal installation, configuration, and operation information.

- *Avaya Secure Router 1000 Installation Guide*

  This guide is designed for network managers and technicians who are responsible for the installation of networking equipment in Telco and service provider network environments.

- *Avaya Secure Router 1000 Command Reference Guide*

  This detailed guide provides a complete alphabetical listing of all non-routing commands including descriptions, syntax, examples, and applicable systems.

- *Avaya Secure Router 1000 Configuration Guide*

  This guide explains how to configure the SR1000 features.

- *Avaya Secure Router 1000 WebUI User Guide*

  This guide explains how to configure the SR1000 using the WebUI.

Guides and release notes are available in PDF format on the Avaya website: http://www.avaya.com.

To view PDF files, Adobe Acrobat® Reader® 4.0 (or later) must be installed on your PC. If you do not have the Adobe Acrobat Reader installed on your system, you can obtain it free from the Adobe website: http://www.adobe.com.

Chapter 2: PROTOCOLS OVERVIEW

BGP4

Border Gateway Protocol Version 4 (also referred to as simply BGP) is an exterior routing protocol used for the global Internet.

Once configured, BGP peers first exchange complete copies of their routing tables (including BGP version, router ID, and keep alive hold time), which are usually very large. Thereafter, only incremental updates (deltas) are sent as changes occur to the routing tables. BGP keeps a current version of the routing table for all peers, keep alive packets are sent to ensure that the connection between BGP peers, and notification packets are sent in response to problems and irregularities. This enables longer running BGP sessions to be more efficient than shorter sessions.

BGP's basic unit of routing information is the BGP path, a route to a certain set of classless interdomain routing prefixes. Paths are tagged with various path attributes, including an autonomous systems (AS) path and next-hop. In fact, one of BGP's most important functions is loop detection at the AS level, using the AS path attribute, which is a list of autonomous systems used for data transport.

The syntax of this attribute is made more complex by its need to support path aggregation when multiple paths are collapsed into one in order to simplify further route advertisements. A more simplified view of an AS path is that it is a list of autonomous systems that a route goes through to reach its destination. Loops are detected and avoided by checking for your own AS number in the AS path's received from neighboring autonomous systems. Every time a BGP path advertisement crosses an AS boundary, the next-hop attribute is changed on the boundary router. Conversely, as a BGP path advertisement is passed among BGP speakers in the same AS, the next-hop attribute is left untouched. Consequently, BGP's next-hop is always the IP address of the first router in the next autonomous system, even though this may actually be several hops away. The AS's interior routing protocol is responsible for computing an interior route to reach the BGP next-hop.

This leads to the distinction between internal BGP (IBGP) sessions (between routers in the same AS) and external BGP (EBGP) sessions (between routers in different AS's). Next-hops are only changed across EBGP sessions, but left intact across IBGP sessions. The two most important consequences of this design are the need for interior routing protocols to reach one hop beyond the AS boundary, and for BGP sessions to be fully meshed within an AS.

Since the next-hop contains the IP address of a router interface in the next AS, and this IP address is used to perform routing, the interior routing protocol must be able to route to this address. This means that interior routing tables must include entries one hop beyond the AS boundary. Furthermore, since BGP does not relay routing traffic from one interior BGP session to another (only from an exterior BGP session to an IBGP session or another EBGP session), BGP speakers must be fully meshed.

RFC Compliance

The following table provides Avaya's BGP RFC compliance information.

Table 2: BGP RFC Compliance

RFC	Description
2385	Protection of BGP sessions via the TCP MD5 signature option
1998	An application of the BGP community attribute in multi-home routing
1997	BGP communities attribute
1775	BGP OSPF interaction
1771	Border Gateway Protocol 4 (BGP-4)

OSPF

Open Shortest Path First (OSPF), a link-state routing protocol, is used for routing IP packets. OSPF offers the following advantages:

- Scalability

  OSPF is designed to operate with larger networks. It does not impose a hop-count restriction and permits its domain to be split into areas for easier management.
- Full subnetting support

  OSPF can fully support subnetting, including Variable Length Subnet Mask (VLSM).
- Tagged routes

  Routes can be tagged with arbitrary values. This eases interoperation with Exterior Gateway Protocols (EGPs), which can tag OSPF routes with AS numbers.
- Meshed networks

  OSPF provides the ability to support complex meshed networks.

The following features are incorporated in Avaya's implementation of OSPF.

- Intra- and inter-area routing
- Broadcast and point-to-point
- Type 1 & Type 2 AS external routes
- Stub areas
- NSSA – Not-So-Stubby-Area
- Route re-distribution
- Authentication – simple & MD5
- RFC 1583 backwards compatibility
- Equal cost multipath
- Configurable routing interface parameters
- Non-intrusive reconfiguration

RFC Compliance

The following table provides Avaya's OSPF RFC compliance information.

Table 3: OSPF RFC Compliance

RFC	Description
2328	OSPF version 2
1587	OSPF NSSA option
1850	OSPF Version 2 Management Information Base

RIP

Routing Information Protocol (RIP) is an interior gateway protocol (IGP), i.e., it routes traffic within a single autonomous system (AS). RIP uses a distance-vector algorithm with hop count as the metric to determine the best route to a destination.

Update messages are sent at configured intervals and when changes occur in the network topology. These messages are used by routers to update their routing tables to maintain currency with the state of the network. When a router updates its routing table, it transmits update messages to other routers in the network to enable them to update their routing tables.

The following list identifies architectural characteristics of RIP:

- The network path is limited to 15 hops. A destination with a greater number of hops is considered unreachable.
- The time and bandwidth requirements will be substantial in a large network
- A fixed metric is used to select routes. Only the best route with the lowest metric is maintained for a specific destination.

The following features are incorporated into Avaya's implementation of RIP:

- RIP v1, v2, and v1 compatibility modes
- Configurable timers
- VLSM
- Split-horizon and split-horizon with poison reverse
- Clear text and MD5 authentication
- Redistribution of connected, static, and OSPF routes
- Redistribution of BGP routes
- Inbound and outbound filtering policies

RFC Compliance

The following table provides Avaya's RIP RFC compliance information.

Table 4: RIP RFC Compliance

RFC	Description
1058	Routing Information Protocol
2453/STD0056	RIP Version 2
1724	RIP Version 2 MIB extension
2082	RIP-II MD5 Authentication

Multicasting

Traditional multicast routing mechanisms such as Distance Vector Multicast Routing Protocol (DVMRP) and Multicast Open Shortest Path First (MOSPF) were intended for use within regions where groups are densely populated or bandwidth is universally plentiful. When groups, and senders to these groups, are distributed sparsely across a wide area, these "dense mode" schemes do not perform efficiently.

Protocol Independent Multicast (PIM)

Protocol Independent Multicast (PIM) protocols route multicast packets to multicast groups. PIM is protocol independent because it can leverage whichever unicast routing protocol is used to populate unicast routing table. There are two modes of PIM protocol – Dense mode (DM) and Sparse mode (SM). Avaya supports SM only.

PIM-DM floods multicast traffic throughout the network initially and then generates prune messages as required. PIM-SM attempts to send multicast data only to networks which have active receivers. This is achieved by having a common Rendezvous Point (RP) known to the senders and receivers and by forming shared trees from the RP to the receivers.

PIM-SM is described in RFC 2362.

Securing Remote Access Using IPSec VPN

These features allow administrators to form a security tunnel to join two private networks over the Internet. The following examples show how to set up an end-to-end tunnel with a single proposal and pre-shared key authentication, with multiple proposals and pre-shared key authentication, and with an SA Bundle, and pre-shared key authentication.

The corporate network no longer has a clearly defined perimeter inside secure building and locked equipment closets. Increasingly, companies have a need to provide remote access to their corporate resources for the employees on the move. Traditionally, remote users could access the corporate LAN through dial-up and ISDN lines which were terminated in the corporate remote access servers. However, these point-to-point connection technologies do not scale well to the growing number of remote users and the corresponding increase in the infrastructure investments and maintenance costs.

A solution to meeting the needs of increasing numbers of remote users and for controlling access costs is to provide remote access through the Internet using firewalls and a Virtual Private Network (VPN). Internet Protocol Security (IPSec) keeps the connection safe from unauthorized users.

In a typical IPSec remote access scenario, the mobile user has connectivity to Internet and an IPSec VPN client loaded on their PC. The remote user connects to the Internet through their Internet service provider and then initiates a VPN connection to the IPSec security gateway (the VPN server) of the corporate office, which is typically an always-on Internet connection.

One of the main limitations in providing remote access is the typical remote user connects with a dynamically assigned IP address provided by the ISP. IPSec uses the IP address of users as an index to apply the Internet Key Exchange (IKE) and IPSec policies to be used for negotiation with each peer. When the VPN client has a dynamic IP address, the VPN server cannot access the policies based on the IP address of the client. Instead, the VPN server uses the identity of the VPN client to access the policies.

Chapter 3: COMMAND LINE INTERFACE

This chapter introduces the Command Line Interface (CLI) hierarchy and the conventions used to describe it. It also introduces the CLI navigation keys and methods, as well as the available help screens.

Command Types

This guide contains two types of commands: transition, or mode change, commands and standard commands.

Transition commands do not affect the Avaya Secure Router 1000 series configuration, they are used to gain access to lower- or next-level commands in the CLI hierarchy. Following each transition command is a brief description, a syntax and usage example, a list of next-level commands, and a list of Secure Routers 1000 series for which the command is applicable.

Note:

In certain instances, transition commands will select an interface for configuration and access next-level commands. For example, the **`configure interface bundle dallas`** command accesses the **`configure interface bundle`** mode and selects or creates the bundle **`dallas`**.

Standard commands are used to configure the Secure Router. Following each standard command is a brief description, a list of parameters and definitions, a syntax and usage example, a list of related commands, and a list of Secure Routers for which the command is applicable.

Context-Sensitive Commands

Some commands are context-sensitive. Once a module, bundle, or Ethernet port has been selected for configuration, all further configuration applies only to the selected interface. Table 5: Context-Sensitive Command Sequence on page 19 shows a context-sensitive command string. In this example, T1 link 1 remains selected for configuration until you exit from the SR/configure/module/t1> prompt.

Table 5: Context-Sensitive Command Sequence

Context-Sensitive Command String		Example
1	Go into the configuration mode.	`SR>configure terminal`

Context-Sensitive Command String		Example
2	Specify the type of interface (T1).	`SR/configure>module t1`
3	Choose the specific interface (T1 link 1).	`SR/configure>module t1 1`
4	From now on, all configuration commands are for T1 link 1 until you exit from module configuration or choose another T1 link.	`SR/configure/module/t1 1>`

Note:

Command strings that require identification of a specific interface are context-sensitive.

Command Conventions

Each command is briefly described and then followed by the complete syntax, which is essentially a map of the command that shows mandatory and optional parameters.

The following tables provide details of the conventions used for syntaxes and examples.

Table 6: Syntax Conventions

For Syntaxes	What it means
normal type	Within syntaxes, "normal type" represents required words that must be entered by the user — except when followed by a parameter setting that is enclosed in angled brackets. In that case, only enter the parameter setting enclosed in the angled brackets. Example 1: Normal type only. In this example, the user enters the word or argument (module) appearing in the syntax in "normal type." Syntax: `module` Command execution: `module` Example 2: Normal type word or argument that is followed by a second normal type word or argument, which is followed by a parameter setting enclosed in angled brackets. In this example, the user enters the first word or argument "connections," appearing in normal type, and then only enters the value "4" of the second word or argument. Syntax:

<table>
<tr><th>For Syntaxes</th><th>What it means</th></tr>
<tr><td></td><td><code>connections connections < n ></code>
Command execution:
<code>connections 4</code>
In other words, the first occurrence of "connections" must be entered because it is not followed by a setting enclosed in angled brackets. The second occurrence of the word "connections" must NOT be entered because it is followed by a setting enclosed in angled brackets. This value of the setting must be entered to execute the command.</td></tr>
<tr><td>[a | b | c]</td><td>Normal brackets "[ ]" indicate optional keywords or arguments.
A vertical bar " | " separates individual settings.
Example:
In this example, the user enters the word "timeout;" must specify either for "tcp" or "udp" for a protocol type; and optionally enters a timeout value "n."
Syntax:
<code>timeout protocol_type < tcp | udp > [seconds < n >]</code>
Command execution:
<code>timeout udp 3600</code></td></tr>
<tr><td>< ></td><td>Angled brackets. All parameter settings are enclosed in angled brackets. The user is directed to choose an appropriate setting. In some cases, the parameter name accompanies the required setting.</td></tr>
<tr><td>[ ]</td><td>Optional parameter settings in each syntax are indicated by normal brackets.</td></tr>
</table>

Abbreviated Commands

You may enter commands by typing the first few characters of each word in a command string. The Secure Router recognizes the unique abbreviated entry and executes the command exactly as if you had entered it fully.

For example, to view the currently running system configuration, you may type **`show configuration running`** at the prompt. You may also type **`sh con ru`** to get the same result. Similarly, you may abbreviate the optional parameter names required by some commands.

For example, a typical entry may be as follows:

```
mlppp mrru 1600 sequence short seg_threshold 1000 differential_delay
100 discriminator 10.1.100.22
```

To save time, you may type the following equivalent abbreviated string:

```
mlppp m 1600 seq short seg 1000 diff 100 dis 10.1.100.22
```

CLI Navigation

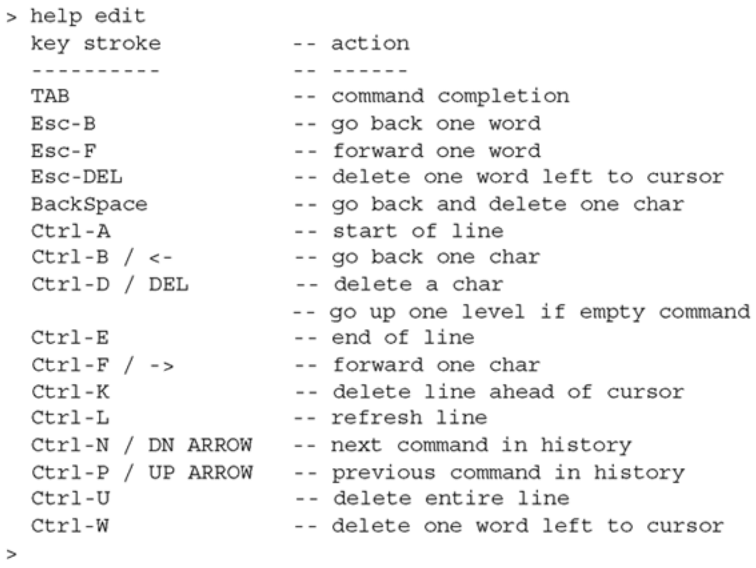
The Tab, Esc, and Ctrl keyboard keys may be used to:

- Move backwards or forwards in the CLI
- Edit entered command strings
- Or accelerate the command entry process

Navigation Keys

You may use the Tab key to quickly enter each word of a command without typing its full name. For example, to enter the **`configure`** command, you may type its first two letters and then press Tab to complete the entire word. Then, you may specify an item to configure by pressing the Spacebar and then pressing Tab repeatedly until the desired sub-command appears. Repeat this sequence for each successive sub-command string until the entire command string appears.

You may also use the other keystrokes shown in Figure 1: Navigation Keys on page 23 during command entry. For example, to back up the cursor without deleting any characters, type `Ctrl-B`. To repeat the last command that you entered, type `Ctrl-P`. To go back several commands, type `Ctrl-P` repeatedly until the desired previous command appears. Or, you may go directly back to the main CLI> prompt from anywhere in the command hierarchy by typing `Ctrl-Z`.



```
> help edit
  key stroke          -- action
  ----------          -- ------
  TAB                 -- command completion
  Esc-B               -- go back one word
  Esc-F               -- forward one word
  Esc-DEL             -- delete one word left to cursor
  BackSpace           -- go back and delete one char
  Ctrl-A              -- start of line
  Ctrl-B / <-         -- go back one char
  Ctrl-D / DEL        -- delete a char
                      -- go up one level if empty command
  Ctrl-E              -- end of line
  Ctrl-F / ->         -- forward one char
  Ctrl-K              -- delete line ahead of cursor
  Ctrl-L              -- refresh line
  Ctrl-N / DN ARROW   -- next command in history
  Ctrl-P / UP ARROW   -- previous command in history
  Ctrl-U              -- delete entire line
  Ctrl-W              -- delete one word left to cursor
>
```

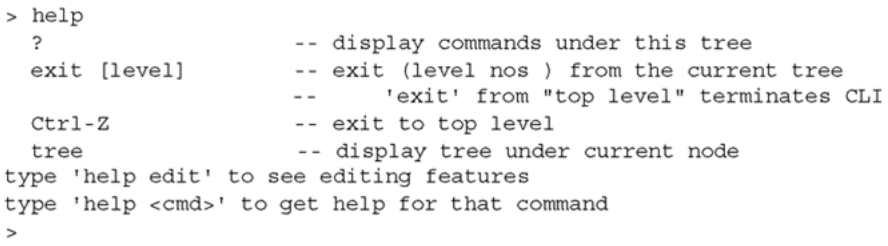
Figure 1: Navigation Keys

Command Help

Command help is available for navigating the CLI command hierarchy and for assistance with specific commands. You may obtain help by using any of the three commands described below.

Help

Type `help` at the main CLI prompt to see the basic Secure Router help information. Or, type `help` followed by a command name to view information about that command. Figure 2: Help Screen on page 23 shows the help screen.

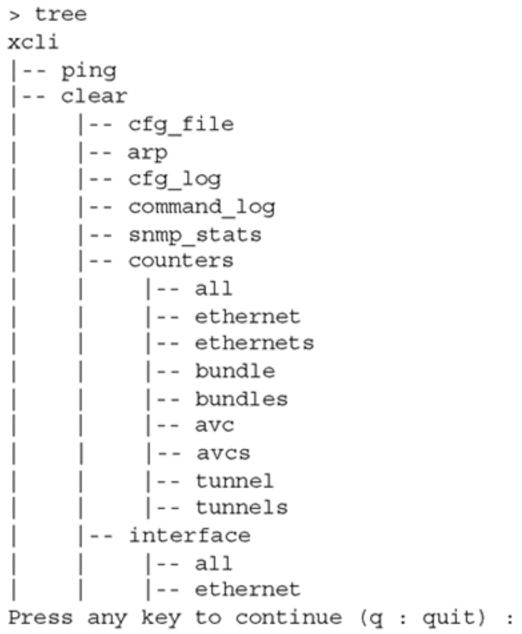


```
> help
  ?                   -- display commands under this tree
  exit [level]        -- exit (level nos ) from the current tree
                      --     'exit' from "top level" terminates CLI
  Ctrl-Z              -- exit to top level
  tree                -- display tree under current node
type 'help edit' to see editing features
type 'help <cmd>' to get help for that command
>
```

Figure 2: Help Screen

Tree

You may view a tree that shows all CLI commands, or a tree that shows only the commands associated with the current command mode (or the routing mode for example). Figure 3: Secure Router CLI Command Tree on page 24 shows two command tree examples. If you type `tree` at the main (SR> or equivalent) prompt, the entire list of Secure Router commands appears. If you type `tree` within a command mode, such as `SR/clear> tree`, the commands associated with this command mode are displayed.

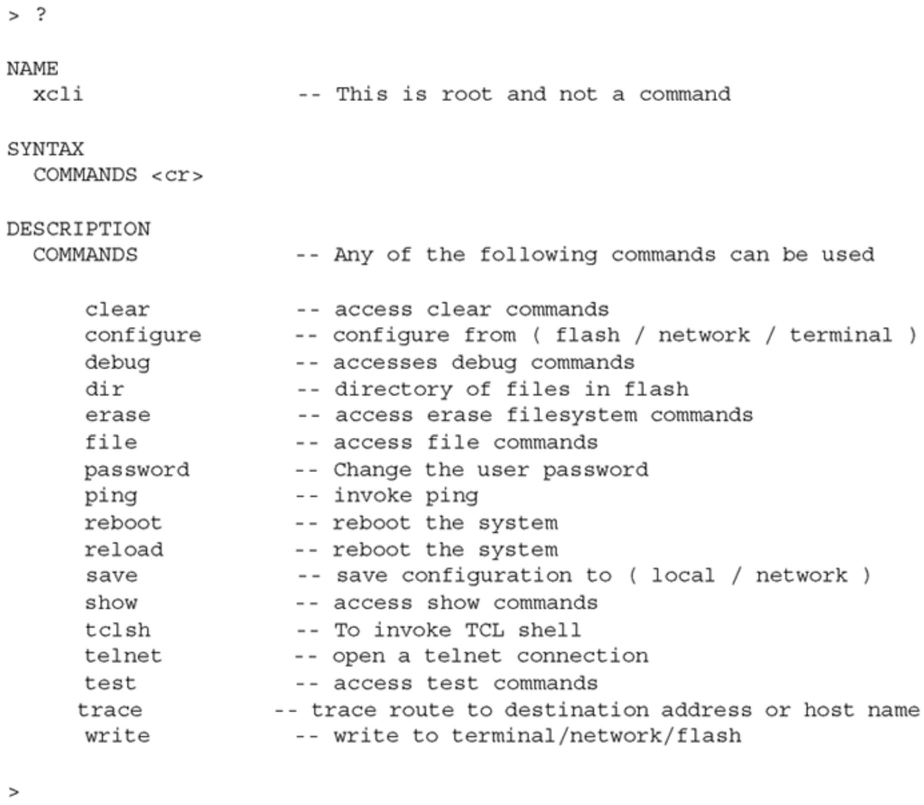


```
> tree
xcli
|-- ping
|-- clear
|     |-- cfg_file
|     |-- arp
|     |-- cfg_log
|     |-- command_log
|     |-- snmp_stats
|     |-- counters
|     |     |-- all
|     |     |-- ethernet
|     |     |-- ethernets
|     |     |-- bundle
|     |     |-- bundles
|     |     |-- avc
|     |     |-- avcs
|     |     |-- tunnel
|     |     |-- tunnels
|     |-- interface
|     |     |-- all
|     |     |-- ethernet
Press any key to continue (q : quit) :
```

Figure 3: Secure Router CLI Command Tree

Question Mark Help Screen

To view help information for a command category, specific command, or a parameter, type the associated word followed by a space and a question mark (?). For example, if you type a question mark at the main command prompt, the Secure Router command categories appear. Figure 4: ? Help Screen on page 25 shows a display of these top-level commands.



```
> ?

NAME
  xcli                -- This is root and not a command

SYNTAX
  COMMANDS <cr>

DESCRIPTION
  COMMANDS            -- Any of the following commands can be used

      clear           -- access clear commands
      configure       -- configure from ( flash / network / terminal )
      debug           -- accesses debug commands
      dir             -- directory of files in flash
      erase           -- access erase filesystem commands
      file            -- access file commands
      password        -- Change the user password
      ping            -- invoke ping
      reboot          -- reboot the system
      reload          -- reboot the system
      save            -- save configuration to ( local / network )
      show            -- access show commands
      tclsh           -- To invoke TCL shell
      telnet          -- open a telnet connection
      test            -- access test commands
     trace          -- trace route to destination address or host name
      write           -- write to terminal/network/flash

>
```

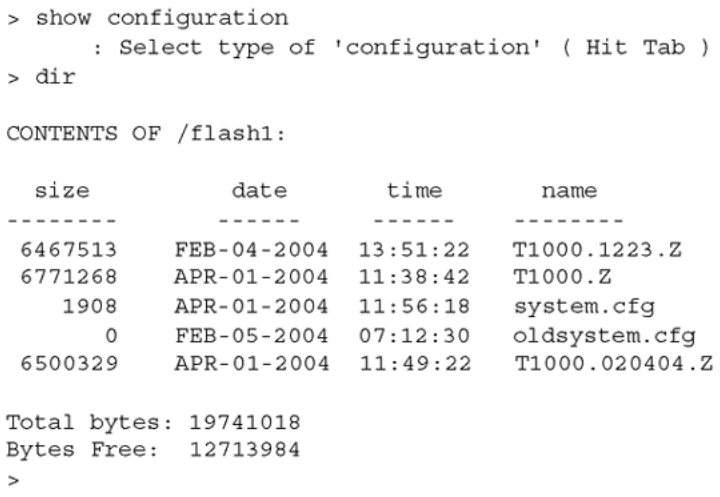
Figure 4: ? Help Screen

> **Note:**
> The default parameters for specific commands appear in parenthesis.

Global Commands

All **show**, **ping**, and **save** commands are available from any level of the CLI. For example, the global **show** commands allow the user to view current configuration settings, alarms, or tests without exiting the configure mode. In Figure 5: Global Show Command on page 26, a user has displayed a bundle summary while configuring a new bundle.

Similarly, the **ping** and **save** commands are available at any level of the CLI command. The **ping** command verifies connectivity between the Secure Router and other network hosts; access to the **save** commands from anywhere in the CLI ensures that your configurations may be saved periodically.



```
> show configuration
      : Select type of 'configuration' ( Hit Tab )
> dir

CONTENTS OF /flash1:

  size          date        time        name
--------       ------      ------     --------
 6467513    FEB-04-2004  13:51:22   T1000.1223.Z
 6771268    APR-01-2004  11:38:42   T1000.Z
    1908    APR-01-2004  11:56:18   system.cfg
       0    FEB-05-2004  07:12:30   oldsystem.cfg
 6500329    APR-01-2004  11:49:22   T1000.020404.Z

Total bytes: 19741018
Bytes Free:  12713984
>
```

Figure 5: Global Show Command

Note:

The tab completion feature is not currently available for global commands.

Chapter 4: POLICY COMMANDS

This chapter provides information about routing policy commands that are supported by Avaya.

configure policy

This command provides access to the next-level commands.

next-level commands
configure policy as_path
configure policy community_list
configure policy ip_access_list
configure policy route_map

configure policy as_path

This command configures the autonomous system path filter for BGP.

AS path access lists are used for matching the AS path attribute in a BGP route. An AS path access list succeeds if any "permit" line in the list matches, or fails if any "deny" line matches. Matching proceeds sequentially and stops at the first match.

The regular expression parameter is an as path regular expression. (For regular expression syntax, see AS PATH REGULAR EXPRESSIONS on page 121.) Note that the regular expression must be enclosed in quotation marks. The AS number is the smallest element of a Secure Router regular expression. It is an integer ranging from 0 to 65535; the Secure Router regular expression matcher is AS number-based.

Any number of AS path access list lines may be declared. They are evaluated in the order declared. If neither permit nor deny is specified, the default is "permit."

parameter	definition
access_list	Access list number Range is 1 - 199.
number	Sequence to insert or delete from an existing AS path entry.

parameter	definition
	Range is 0 - 65535.
action	
deny	Deny AS path.
permit	Permit AS path.
regular_expression	Regular expression to match the AS paths. Enter a quoted string. Refer to AS PATH REGULAR EXPRESSIONS on page 121 for more information about regular expressions.

syntax:

```
[ no ] policy as_path access_list < n > number < n > action < deny |
permit > regular_expression < "string" >
```

example:

```
SR/configure> policy as_path 1 120 permit "100"
```

example:

```
SR/configure> policy as_path 1 121 deny ".* 101 .*"
```

applicable systems:

All models.

configure policy community_list

This command accesses next-level commands for adding extended or standard community lists.

Community lists are used for matching the "community" attribute in a BGP route. A community list succeeds if any "permit" line in the list matches, or fails if any "deny" line matches. Matching proceeds sequentially and stops at the first match. A line in a community list is normally said to match if the route being tested contains at least all of the communities listed in the line. That is, it may contain additional communities as well. If the exact-match keyword is used, then it must contain exactly the same communities as listed.

The communities parameter can be:

- local_as
- no_advertise
- no_export

- aa_nn (an integer between 0 and 65,535)
- community (an integer between 1 and 4294967295

Note that "exact_match" is supported in the community_list as well as at the route_map level. If neither permit nor deny is specified, the default is permit. If no community is specified, any route will be matched, regardless of what communities are present. The route will even be matched if the community path attribute is not present. Any number of community list lines may be declared. They are evaluated in the order declared.

next-level commands
configure policy community_list extended_community
configure policy community_list standard_community

configure policy community_list extended_community

This command configures an extended community list as part of the policy.

parameter	definition
community_list	Extended community list number The range is 100 - 199.
community_index	Community index number The range is 0 - 65535.
action	
deny	Specify a community to reject.
permit	Specify a community to permit.
community	A list of community numbers The range is 1 - 4294967295. This list can contain a maximum of 32 numbers.
generate_local_as	
local_as	Do not send out local AS.
aa_nn	Community number in aa_nn format This list can contain a maximum of 32 numbers.
generate_no_advertise	
no_advertise	Do not advertise to any neighbor.
generate_no_export	
no_export	Do not send to next AS

syntax:

```
[ no ] policy community_list extended_community community_list < n >
community_index < n > action < deny | permit > [ community < n > ]
[ generate _local_as < local_as > ] [ aa_nn < n > ]
[ generate_no_advertise < no_advertise > ] [ generate_no_export <
no_export > ]
```

example:

```
SR/configure> policy community_list extended_community 100 1 deny
community 44 45 local_as aa_nn 400:500 no_advertise
```

applicable systems:

All models.

configure policy community_list standard_community

This command configures a standard community list as part of the routing policy.

parameter	definition
community_list	Extended community list number The range is 100 - 199.
community_index	Community index number The range is 0 - 65535.
action	
deny	Specify a community to reject.
permit	Specify a community to permit.
community	A list of community numbers The range is 1 - 4294967295. This list can contain a maximum of 32 numbers.
generate_local_as	
local_as	Do not send out local AS.
aa_nn	Community number in aa_nn format This list can contain a maximum of 32 numbers.
generate_no_advertise	
no_advertise	Do not advertise to any neighbor.
generate_no_export	
no_export	Do not send to next AS

syntax:

```
[ no ] policy community_list standard_community community_list < n >
community_index < n > action < deny | permit > [ community < n > ]
[ generate_local_as < local_as > ] [ aa_nn < n > ]
[ generate_no_advertise <no_advertise > ] [ generate_no_export <
no_export > ]
```

example:

```
SR/configure> policy community_list standard_community 90 150 permit
community 40 45 local_as aa_nn 655:232592 no_advertise
```

example:

```
SR/configure/policy> community_list standard_community 90 150 permit
community 42949672 no_advertise
```

applicable systems:

All models.

configure policy ip_access_list

This command configures the IP access list for routes.

Ip access lists are used for matching any type of route prefix. An IP access list is said to succeed if any "permit" line in the list matches, or fails, if any "deny" line matches. Matching proceeds sequentially and stops at the first match. A line in an IP access list is said to match according to the rules listed below.

- network netmask

  Matches addresses as follows: The bits in the address part of the route being masked that are not covered by "one" bits in net mask must be equal to the corresponding bits in networkt. The "one" bits in net mask are sometimes referred to as "don't care" bits, because the policy engine does not care what their values are.

- network netmask mask maskmask

  Matches addresses as follows: The first pair of parameters (network, maskmask) match the address part of the route just as in the previous (network netmask) form. The second pair of parameters (mask, jaskmask) are used to match against the mask part of the route being matched in a similar fashion. That is, the route is matched if the address part matches and the bits in the mask that are not covered by "one" bits in net mask are equal to the corresponding bits in mask.

If neither permit nor deny is specified, the default is permit. All kinds of access_list entries may be mixed freely within a list, and there are no restrictions on what the access_list number may be. Any number of IP access list lines may be declared. They are evaluated in the order declared.

parameter	definition
access_list	Access list number The range is 1 - 99
number	Sequence to insert to or delete from an existing access list entry. The range is 0 - 65535.
action	
deny	Route map deny set operation.
permit	Route map permit set operation.
network	Network route (IP address in dotted notation)
netmask	Network mask as wildcard bits (IP address in dotted notation)
mask	Network route's mask (IP address in dotted notation)
maskmask	Wildcard mask for network route's mask (in dotted notation)

syntax:

```
[ no ] policy ip_access_list access_list < n > number < n > action <
deny | permit > [ network < IP address> ] [ netmask < IP address > ]
[ mask < IP address > ] [ maskmask < IP address > ]
```

example:

```
SR/configure> policy ip_access_list 1 1 permit network 10.0.0.0
netmask 0.255.255.255
```

This example permits prefixes 10.0.0.0/8, 10.0.0.0/9 and so on.

example:

```
SR/configure> policy ip_access_list 1 1 permit network 10.0.0.0
netmask 0.255.255.255 mask 255.0.0.0 maskmask 0.255.255.255
```

This example restricts the prefixes to 10.0.0.0/8 only.

applicable systems:

All models.

configure policy route_map

This command configures the policy for router route maps.

Route maps are used for general-purpose matching of routes and setting of route attributes. Each route_map is comprised of one or more route_map clauses, of the form shown below.

route_map name number [permit | deny]

Example

match statements set statements

A route_map clause is said to match if each of its match statements matches, according to the rules given below. A route_map is said to succeed if one of its permit clauses clauses matches, and fails if one of its deny clauses matches. Matching proceeds sequentially and stops at the first match. If the route_map succeeds, the actions specified by the set statements in the matched clause are performed.

If neither permit nor deny is specified, the default is permit.

Match statements can be:

- match as_path
- match community
- match ip ip_address

Set statements can be:

- set as_path
- set community
- set local_preference
- set metric
- set origin
- set distance
- set metric_type

parameter	definition
name	Route map name
number	A sequence to insert to or delete from exiting route map. The range is 0 - 65535.
action	
deny	Deny the route map. This is the default value.
permit	Permit the route map.

syntax:

```
[ no ] policy route_map name number [ action < deny | permit > ]
```

example:

```
SR/configure> policy route_map Block100 1 permit
```

next-level commands
configure policy route_map commit
configure policy route_map match
configure policy route_map set

applicable systems:

All models.

configure policy route_map match

This command accesses next-level commands for configuring the policy for matching parameters of the routes.

next-level commands
configure policy route_map match as_path
configure policy route_map match community
configure policy route_map match ip

configure policy route_map match as_path

This command matches any of the specified BGP AS path access lists.

parameter	definition
path_list	AS path access list The range is 1 - 199; the maximum list size is 32.

syntax:

```
[ no ] policy match as_path path_list < n >
```

example:

```
SR/configure>/policy/route_map Block100 1> match as_path 1
```

next-level commands
configure policy route_map match ip
configure policy route_map match community

applicable systems:

All models.

configure policy route_map match community

This command matches any of the specified BGP community lists.

syntax:

```
[ no ] policy match community
```

example:

```
SR/configure/policy/route_map Block100 1> match community
```

next-level commands
configure policy route_map match as_path
configure policy route_map match ip

applicable systems:

All models.

configure policy route_map match ip ip_address

This command distributes routes matching the prefix against any of the specified IP access lists.

parameter	definition
ip_list	Ip access list number(s) Enter a list of numbers. The range is 1 - 199. A maximum of 32 numbers can be in the list.

syntax:

```
[ no ] match ip ip_address ip_list < n >
```

example:

```
SR/configure/policy/route_map Block100 1> match ip ip_address 20
```

applicable systems:

All models.

configure policy route_map match source-protocol

This command matches the source protocol

parameter	definition
match_bgp	bgp protocol routes This parameter may have the following values: bgp
match_ospf	osfp protocol routes This parameter may have the following values: ospf
match_rip	rip protocol routes This parameter may have the following values: rip
match_static	static routes This parameter may have the following values: static
match_connected	connected routes This parameter may have the following values: connected

syntax:

```
source-protocol [ match_bgp ] [ match_ospf ] [ match_rip ]
[ match_static ] [ match_connected ]
```

example:

```
host/configure/policy/route_map test 1 > match source-protocol bgp
ospf rip
```

applicable systems:

All models.

configure policy route_map set

This command provides access to next-level commands to set parameters for the routes.

next-level commands
configure policy route_map set as_path
configure policy route_map set community
configure policy route_map set distance
configure policy route_map set local_preference
configure policy route_map set metric
configure policy route_map set metric_type
configure policy route_map set origin

configure policy route_map set as_path

This command configures a character string for a BGP AS-path attribute.

parameter	definition
prepend	AS path access list Enter a list of numbers. The range is 1 - 65535; the maximum list size is 32.
tag	Set tag as an AS path attribute. Enter a number.

syntax:

```
[ no ] set as_path [ prepend < n > ] [ tag < n > ]
```

example:

```
SR/configure/policy/route_map Block100 1> set as_path prepend 100 250
tag 0
```

related commands
configure policy route_map set community
configure policy route_map set distance

related commands
configure policy route_map set local_preference
configure policy route_map set metric
configure policy route_map set metric_type
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set community

This command configures the policy for community attributes.

Set the community attribute to the given value or list of values. If the additive keyword is specified, the list of values augments any communities already present. If the additive keyword is not specified, the list of values overwrites any communities already present.

parameter	definition generate_
number	Community number (unsigned) The range is 1 - 4294967294 The maximum numbers in the list is 32.
aa_nn	Community number in aa_nn format Enter a number or a list of numbers separated by spaces. The maximum numbers in the list is 32
generate_additive	
additive	Add to the existing community.
generate_local_as	
local_as	Do not send outside local AS.
generate_no_advertise	
no_advertise	Do not advertise to any neighbor.
generate_no_export	
no_export	Do not send to next AS

syntax:

```
[ no ] set community number [ < n > ] [aa_nn < n > ]
[ generate_additive < additive > ] [ generate_local_as < local_as > ]
[ generate_no_advertise < no_advertise > ] [ generate_no_export <
no_export > ]
```

example:

```
SR/configure/policy/route_map Block100 1> set community aa_nn 500:60
```

related commands
configure policy route_map set as_path
configure policy route_map set distance
configure policy route_map set local_preference
configure policy route_map set metric
configure policy route_map set metric_type
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set distance

This command sets the BGP protocol preference for the path attribute.

parameter	definition
distance	Default preference value The range is 0 - 255.

syntax:

```
[ no ] set distance distance < n >
```

example:

```
SR/configure/policy/route_map Block100 1> set distance 20
```

related commands
configure policy route_map set as_path
configure policy route_map set community
configure policy route_map set local_preference
configure policy route_map set metric
configure policy route_map set metric_type

related commands
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set local_preference

This command configures the BGP local preference path attribute.

parameter	definition
local_preference	Preference value The range is 1 - 4292967294.

syntax:

```
[ no ] set local_preference local_preference < n >
```

example:

```
SR/configure/policy/route_map Block100 1> set local_preference 50
```

related commands
configure policy route_map set as_path
configure policy route_map set community
configure policy route_map set distance
configure policy route_map set metric
configure policy route_map set metric_type
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set metric

This command configures the metric value for the destination routing protocol.

parameter	definition
metric	Metric value The range is 1 - 4294967294.

syntax:

```
[ no ] set metric metric < n >
```

example:

```
SR/configure/policy/route_map Block100 1> set metric 120
```

related commands
configure policy route_map set as_path
configure policy route_map set community
configure policy route_map set distance
configure policy route_map set local_preference
configure policy route_map set metric_type
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set metric_type

This command configures the metric type for a route.

parameter	definition
type	Internal
internal	Use the IGP metric as the MED for BGP.

syntax:

```
[ no ] set metric_type type < internal >
```

example:

```
SR/configure/policy/route_map Block100 1> set metric_type internal
```

related commands
configure policy route_map set as_path

related commands
configure policy route_map set community
configure policy route_map set distance
configure policy route_map set local_preference
configure policy route_map set metric
configure policy route_map set origin

applicable systems:

All models.

configure policy route_map set origin

This command configures the origin value for the BGP route.

parameter	definition
origin	
egp	EGP protocol
igp	IGP protocol
incomplete	Unknown protocol type

syntax:

```
[ no ] set origin origin < egp | igp | incomplete >
```

example:

```
SR/configure/policy/route_map Block100 1> set origin igp
```

applicable systems:

All models.

next-level commands
configure policy route_map set origin egp
configure policy route_map set origin igp
configure policy route_map set origin incomplete

Chapter 5: GENERIC ROUTING COMMANDS

This chapter contains routing commands that are not protocol specific. These commands can be used interchangeably with the three routing protocols supported by Avaya.

configure router

This command provides access to next-level commands.

next-level commands
configure router routerid

configure router routerid

This command configures a router for routing operation.

syntax:

```
[ no ] router routerid < IP address>
```

example:

```
SR/configure> router routerid 10.10.10.10
```

applicable systems:

All models.

Important:

Please restart OSPF for the router-id to take effect if the router-id changes dynamically

show ip routes

This command displays IP routing information for Ethernet ports.

parameter		definition
network		Network IP address Enter an IP address.
mask		Network mask Enter a netmask address
protocol		
	all	All protocols
	bgp	Border Gateway protocol (BGP)
	connected	Connected routes
	ospf	Open Shortest Path First protocol (OSPF)
	rip	Routing Information Protocol (RIP)
	static	Static routes
database		
	fib	FIB routes
summary		summary of all routes
options		
	exact	exact match with network/mask
	begin	begin with the route that matches the network/mask wildcard
	include	include routes that match the network/mask wildcard
	exclude	exclude routes that match the network/mask wildcard

syntax:

```
show ip routes [ network < IP address > ] [ mask < netmask > ]
[ protocol < all | bgp | connected | ospf | rip | static > ] [ database
< rib | fib > ]
```

The following table provides parameter definitions for the following screen display examples.

Table 7: Parameter Definitions

term	definition
Network	Indicates the address of the remote network.
Next Hop	Specifies the address of the next router to the remote network
Interface	Specifies the interface through which the specified network can be reached.
PVC >	Virtual (logical) circuit identification number.
Distance	The administrative distance for the route.
Metric	The metric for the route.

By default, information is displayed for all routes in the routing table. To display only specific route information, specify the appropriate protocol or the network mask.

example:

To display all routes, issue the `show ip routes` command.

example:

To display the route for a specific network and subnet, issue the `show ip routes network 123.1.2.0 mask 255.255.255.0` command.

example:

To display the connected ip routes, issue the `show ip routes connected` command.

example:

To display static routes, issue the `show ip routes static` command.

example:

To display RIP routes, issue the `show ip routes rip` command.

example:

To display ospf routes, issue the `show ip routes ospf` command.

example:

```
SR/show> ip routes bgp
```

The following screen display example is a typical display showing the destination IP address, metric, netmask and gateway, status, Ethernet interface, and type of route.

applicable systems:

All models.

Chapter 6: BGP4 CLEAR COMMANDS

Use BGP clear commands to clear bgp configuration settings.

clear ip bgp

This command provides access to the following next-level commands.

syntax:

```
clear ip bgp
```

clear ip bgp

next-level commands
clear ip bgp all
clear ip bgp group
clear ip bgp neighbor

example:

```
SR> clear ip bgp
```

applicable systems:

All models.

clear ip bgp all

This command removes all BGP neighbor connections.

syntax:

```
clear ip bgp all
```

example:

```
SR> clear ip bgp all
```

related commands
clear ip bgp group
clear ip bgp neighbor

applicable systems:

All models.

clear ip bgp group

This command removes all connections for a BGP group.

parameter	definition
group_name	Name of the group

syntax:

```
clear ip bgp group group_name < name >
```

example:

```
SR> clear ip bgp group north
```

In this example, all BGP connections that belong to neighbor group north will be cleared.

related commands
clear ip bgp all
clear ip bgp neighbor

applicable systems:

All models.

clear ip bgp neighbor

This command removes a specified BGP neighbor connection.

parameter	definition
ip_address	The IP address of the neighbor Enter an IP address (in dotted notation) to be cleared.
remote_as	The AS number of the remote neighbor to be cleared. The range is from 1 - 65535.

syntax:

```
clear ip bgp neighbor ip_address < IP address > remote_as < n >
```

example:

```
SR> clear ip bgp neighbor 10.1.1.1 200
```

related commands
clear ip bgp all
clear ip bgp group

applicable systems:

All models.

Chapter 7: BGP4 CONFIGURE COMMANDS

Use BGP configure commands to configure all BGP4 parameters.

configure router bgp

This command configures BGP routing protocol on a router and provides access to the next-level commands listed below.

parameter	definition
as_number	The number of an autonomous system. The range is 1 - 65535.

syntax:

```
[ no ] router bgp as_number
```

example:

```
SR/configure> router bgp <as_number>
```

next-level commands
configure router bgp <as_number> aggregate_address
configure router bgp <as_number> always_compare_med
configure router bgp <as_number> distance
configure router bgp <as_number> default_metric
configure router bgp <as_number> group
configure router bgp <as_number> neighbor
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp aggregate_address

This command is used to aggregate routes.

parameter	definition
network	Network IP address in dotted notation
mask	Network subnet mask address in dotted notation
generate_as_set	
as_set	Generates AS path information Form a verbose aggregate, whose AS path contains a leading AS sequence representing the common leading sequence of all contributing routes, and whose AS path contains a trailing AS set representing all ASes in all contributing paths that could not be included in the AS sequence. By default, this feature is off, and the AS path is truncated when the aggregate is formed.
generate_summary_only	
summary_only	Filters more specific routes from updates Suppresses transmission of any contributing routes if an aggregate exists. Note that the contributing route will not be sent even if an outgoing route_map blocks the sending of the aggregate itself. This cannot be combined with the suppress_map parameter.
suppress_map	Name of the route map to suppress Uses the named route_map to suppress the transmission of selected contributing routes. Contributing routes that do not match the route_map will not be suppressed. This cannot be combined with the summary_only parameter.
advertise_map	Name of route map to control attribute advertisement Selects the routes that contribute to the aggregate. The aggregate will only be formed if matching routes exist. Only the matching routes will be suppressed if summary_only or suppress_map are configured.
attribute_map	Name of route map for setting attributes Specifies attributes to be set on the aggregate when it is transmitted.

syntax:

```
[ no ] aggregate_address < IP address > < subnet mask >
[ generate_as_set < as_set > ] [ generate_summary_only < summary_only
> ] [ suppress_map < name > ] [advertise_map < name > ]
[ attribute_map < name > ]
```

example:

```
SR/configure/router/bgp 10> aggregate_address 100.3.0.0 255.255.0.0
```

related commands
configure router bgp <as_number> always_compare_med
configure router bgp <as_number> distance
configure router bgp <as_number> default_metric
configure router bgp <as_number> group
configure router bgp <as_number> neighbor
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp always_compare_med

This command configures a router to allow the comparison of the multi-exit discriminator for paths from neighbors in different autonomous systems.

Normally, MED comparison is done on paths within the same autonomous system. This command allows the comparison to be made for paths received from other autonomous systems.

syntax:

```
[ no ] always_compare_med
```

example:

```
SR/configure/router/bgp 10> always_compare_med
```

related commands
configure router bag <as_number> aggregate_address
configure router bgp <as_number> distance
configure router bgp <as_number> default_metric
configure router bgp <as_number> group
configure router bgp <as_number> neighbor

related commands
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp default_metric

This command configures the default metric value for redistributed BGP routes.

This command forces the routing protocol to use the same metric value for all redistributed routes.

parameter	definition
default_metric	The default metric value. The range is 1 - 4294967294.

syntax:

```
[ no ] default_metric < n >
```

example:

```
SR/configure/router/bgp 10>default_metric 2000
```

related commands
configure router bgp <as_number> aggregate_address
configure router bgp <as_number> always_compare_med
configure router bgp <as_number> distance
configure router bgp <as_number> group
configure router bgp <as_number> neighbor
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp distance

This command changes the default distance value on a router.

Higher values are preferred.

parameter	definition
distance	Default preference value The range is 0-255; the default is 170.

syntax:

```
[ no ] distance distance < n >
```

example:

SR/configure/router/bgp 10> distance 20

Table 8: Default Route Preference (Administrative Distance) Values

How Route is Learned	Default Preference	Command to Modify Default Preference
Directly connected network	0	Not configurable.
Static	1	Not configurable.
OSPF non-external route	10	configure router ospf distance ospf non_external
RIP	100	configure router rip distance
Generated or aggregate	130	Applicable to BGP only, and is not configurable.
OSPF AS external routes	150	configure router ospf distance ospf external
BGP	170	configure router bgp distance

related commands
configure router bgp <as_number> aggregate_address
configure router bgp <as_number> always_compare_med
configure router bgp <as_number> default_metric
configure router bgp <as_number> group
configure router bgp <as_number> neighbor
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp egbp_ecmp

This command configures the BGP EBGP_ECMP

syntax:

```
SR-3120/configure > router bgp 3
```

example:

```
SR-3120/configure/router/bgp 3 > ebgp_ecmp
```

applicable systems:

All models.

configure router bgp group

This command configures BGP groups

Neighbors with the same update policies are more easily managed when they are in groups. Group organization simplifies configuration and streamlines the update process. Neighbor group members inherit all configuration options of a group. The BGP group sub commands are similar to those found under the neighbor tree, but they are applied to all neighbors in the group.

parameter	definition
name	Group name to be configured
group_type	
external	External routing group Default group name = DEFAULT-EXTERNAL
external_rt	External routing group Default group name = DEFAULT-EXTERNAL_RT
internal	Internal routing group Default group name = DEFAULT-INTERNAL

syntax:

```
[ no ] group name group_type
```

example:

```
SR/configure/router/bgp 10> group toronto internal
```

applicable systems:

All models

configure router bgp group

This command configures BGP groups.

Neighbors with the same update policies are more easily managed when they are in groups. Group organization simplifies configuration and streamlines the update process. Neighbor group members inherit all configuration options of a group. The BGP group sub commands are similar to those found under the neighbor tree, but they are applied to all neighbors in the group.

parameter	definition
name	Group name to be configured
group_type	
external	External routing group Default group name = AvayaBgpExternal
external_rt	External routing group Default group name = AvayaBgpExternalRt
internal	Internal routing group Default group name = AvayaBgpInternal

syntax:

```
[ no ] group name < name > group_type < external | external_rt |
internal | internal_ rt >
```

example:

```
SR/configure/router/bgp 10> group toronto internal
```

next-level commands
configure router bgp group distribute_list
configure router bgp group filter_list
configure router bgp group next_hop_self
configure router bgp group password

next-level commands
configure router bgp group remove_private_AS
configure router bgp group route_map

applicable systems:

All models.

configure router bgp group distribute_list

This command configures filter updates to this group.

parameter	definition
access_list	IP access list number The range is 1-199.
filter_option	
out	Outbound direction

syntax:

```
[ no ] distribute_list access_list < n > filter_option < out >
```

example:

```
SR/configure/router/bgp 10/group toronto internal>
```

related commands
configure router bgp <as_number> group filter_list
configure router bgp <as_number> group next_hop_self
configure router bgp <as_number> group password
configure router bgp <as_number> group remove_private_AS
configure router bgp <as_number> group route_map

applicable systems:

All models.

configure router bgp group filter_list

This command configures BGP filters for a specified group.

parameter	definition
access list	AS path access list The range is 1-199.
filter_option	
out	Outbound direction

syntax:

```
[ no ] filter_list access list < n > filter_option < out >
```

example:

```
SR/configure/router/bgp 10/group toronto internal> filter_list 103
out
```

related commands
configure router bgp <as_number> group distribute_list
configure router bgp <as_number> group next_hop_self
configure router bgp <as_number> group password
configure router bgp <as_number> group remove_private_AS
configure router bgp <as_number> group route_map

applicable systems:

All models.

configure router bgp group next_hop_self

This command disables the next hop calculation for all peers in the group.

syntax:

```
next_hop_self
```

example:

```
SR/configure/router/bgp 10/group blue external> next_hop_self
```

related commands
configure router bgp <as_number> group distribute_list
configure router bgp <as_number> group filter_list
configure router bgp <as_number> group password
configure router bgp <as_number> group remove_private_AS
configure router bgp <as_number> group route_map

applicable systems:

All models.

configure router bgp group password

This command configures the TCP MD5 password to enable MD5 authentication for a BGP group.

parameter	definition
md5_password	TCP MD5 password (string) for the group Enter a word.

syntax:

```
[ no ] password md5_password < string >
```

example:

```
SR/configure/router/bgp 10/group toronto internal> password rt56htd
```

related commands
configure router bgp <as_number> group distribute_list
configure router bgp <as_number> group filter_list
configure router bgp <as_number> group next_hop_self
configure router bgp <as_number> group remove_private_AS
configure router bgp <as_number> group route_map

applicable systems:

All models.

configure router bgp group remove_private_AS

This command removes the private AS number from updates that are sent out.

syntax:

```
[ no ] remove_private_AS
```

example:

```
SR/configure/router/bgp 10/group toronto internal> remove_private_AS
```

related commands
configure router bgp <as_number> group distribute_list
configure router bgp <as_number> group filter_list
configure router bgp <as_number> group next_hop_self
configure router bgp <as_number> group password
configure router bgp <as_number> group route_map

applicable systems:

All models.

configure router bgp group route_map

This command configures a route map to a BGP group.

This command can only be applied in the outbound direction.

parameter	definition
route_map	Route map name
route_map_options	
out	Outbound direction

syntax:

```
[ no ] route_map route_map < name > route_map_options < out >
```

example:

```
SR/configure/router bgp 10/group toronto internal> route_map foo out
```

related commands
configure router bgp <as_number> group distribute_list
configure router bgp <as_number> group filter_list
configure router bgp <as_number> group next_hop_self
configure router bgp <as_number>group password
configure router bgp <as_number> group remove_private_AS

applicable systems:

All models.

configure router bgp neighbor

This command configures a BGP neighbor.

parameter	definition
IP address	The IP address of the neighbor in dotted notation
remote_as	The AS number The range is 1 - 65535.

syntax:

```
[ no ] neighbor IP address < IP address > remote_as < n >
```

example:

```
SR/configure/router/bgp 10> neighbor 101.101.1.2 4
```

next-level commands
configure router bgp <as_number> neighbor <ip address> <remote_as> advertisement_interval
configure router bgp <as_number> neighbor <ip address> <remote_as> allowbadid
configure router bgp <as_number> neighbor <ip address> <remote_as> default_originate
configure router bgp <as_number> neighbor <ip address> <remote_as> description
configure router bgp <as_number> neighbor <ip address> <remote_as> distribute_list
configure router bgp <as_number> neighbor <ip address> <remote_as> ebgp_multihop
configure router bgp <as_number> neighbor <ip address> <remote_as> filter_list
configure router bgp <as_number>neighbor <ip address> <remote_as> ignore_first_as

next-level commands
configure router bgp <as_number> neighbor <ip address> <remote_as> keep
configure router bgp <as_number> neighbor <ip address> <remote_as> logupdown
configure router bgp <as_number> neighbor <ip address> <remote_as> maximum_prefix
configure router bgp <as_number> neighbor <ip address> <remote_as> neighbor_group
configure router bgp <as_number> neighbor <ip address> <remote_as> password
configure router bgp <as_number> neighbor <ip address> <remote_as> route_map
configure router bgp <as_number> neighbor <ip address> <remote_as> timers
configure router bgp <as_number> neighbor <ip address> <remote_as> update_source

related commands
configure router bgp <as_number> aggregate_address
configure router bgp <as_number> always_compare_med
configure router bgp <as_number> distance
configure router bgp <as_number>default_metric configure router bgp <as_number> ecmp
configure router bgp <as_number> group
configure router bgp <as_number> redistribute

applicable systems:

All models.

configure router bgp neighbor advertisement_interval

This command configures the minimum time interval for sending BGP route updates.

parameter	definition
advertisement_interval	Time, in seconds The range is 1 - 600 seconds.

syntax:

```
[ no ] advertisement_interval advertisement_interval < n >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4>
advertisement_interval 60
```

applicable systems:

All models.

configure router bgp neighbor allowbadid

This command permits BGP sessions to be established with routers that represent their router ID as 0.0.0.0 or 255.255.255.255.

syntax:

```
[ no ] allowbadid
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> allowbadid
```

applicable systems:

All models.

configure router bgp neighbor default_originate

This command sends the default route to the neighbor.

parameter	definition
route_map	The name of the route map

syntax:

```
[ no ] default_originate [ route_map < name > ]
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> default_originate
altmap5
```

applicable systems:

All models.

configure router bgp neighbor description

This command describes or identifies a neighbor router.

parameter	definition
neighbor_description	Text string in quotes describing neighbor

syntax:

```
[ no ] description neighbor_description < "string" >
```

example:

SR/configure/router/bgp 10/neighbor 101.101.1.2 4> description "foo1"

applicable systems:

All models.

configure router bgp neighbor distribute_list

This command configures filter updates to or from this neighbor.

parameter	definition
access_list	The IP access list number. The range is 1 - 199.
filter_option	
in	Inbound filter list

syntax:

```
[ no ] distribute_list access_list < n > filter_option < in >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> distribute_list
101 in
```

applicable systems:

All models.

configure router bgp neighbor ebgp_multihop

This command configures multihop EBGP on a neighbor.

syntax:

```
[ no ] ebgp_multihop
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> ebgp_multihop
```

applicable systems:

All models.

configure router bgp neighbor filter_list

This command configures BGP filters.

parameter	definition
access_list	AS path access list The range is 1 - 199.
access_list_option	
in	Inbound filter list

syntax:

```
[ no ] filter_list access_list < n > access_list_option < in >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> filter_list 103 in
```

applicable systems:

All models.

configure router bgp ignore_first_as

This command is used to ignore unconfigured first AS

syntax:

```
ignore_first_as
```

example:

```
SR/configure/router/bgp 10/neighbor 10.10.10.1 10 > ignore_first_as
```

applicable systems:

all models

configure router bgp neighbor keep

This command configures neighbor route storage options.

parameter	definition
keep_option	
all	Keep all non-active routes
none	Don't store non-active routes

syntax:

```
keep keep_option < all | none >
```

example:

```
SR/configure/router/bgp 10/neighbor 10.10.20.1 2> keep all
```

applicable systems:

All models.

configure router bgp neighbor logupdown

This command configures logging of established state transition changes of a neighbor.

syntax:

```
[ no ] logupdown
```

example:

```
SR/configure/router/bgp10/neighbor 101.101.1.2 4> logupdown
```

applicable systems:

All models.

configure router bgp neighbor maximum_prefix

This command configures the maximum number of BGP routes to be accepted.

If the neighbor sends more prefixes than are configured, the connection to this neighbor will be broken.

parameter	definition
prefix_number	Maximum prefix limit The range is 1 - 1000000.

syntax:

```
maximum_prefix prefix_number < n >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> maximum_prefix
100000
```

applicable systems:

All models.

configure router bgp neighbor neighbor_group

This command configures a neighbor to a specific group.

parameter	definition
neighbor_group	The name of a neighbor group.

syntax:

```
[ no ] neighbor_group neighbor_group < name >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> neighbor_group
internal-group
```

applicable systems:

All models.

configure router bgp neighbor password

This command configures a password for md5 authentication.

parameter	definition
md5_password	TCP MD5 password for the BGP session Enter a word (maximum 80 characters).

syntax:

```
md5_password < string >
```

example:

```
SR/configure/router/bgp 10/neighbor 10.10.20.1 2> md5_password asdf
```

applicable systems:

All models.

configure router bgp neighbor route_map

This command applies a route map to a neighbor.

A similar command exists under the group tree for applying route_map to a group of neighbors in the outbound direction.

parameter	definition
route_map	The name of a route map
route_map_options	Filter options
in	Inbound direction

syntax:

```
[ no ] route_map route_map < name > route_map_options < in >
```

example:

```
SR/configure/router/bgp 10/neighbor 100.50.23.3 4> route_map B01 in
```

applicable systems:

All models.

configure router bgp neighbor timers

This command configure keepalive timers for a neighbor (peer).

The holdtime timer value is calculated as three times the value of the keepalive timer.

parameter	definition
keepalive	The keepalive interval The range is 2 - 21845; the default is 60.

syntax:

```
[ no ] timers keepalive < n >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> timers 120
```

applicable systems:

All models.

configure router bgp neighbor update_source

This command configures the source of BGP TCP connections for a specified neighbor as the IP address specified, instead of the IP address of a physical interface.

This address will be used as the source address for routing updates.

syntax:

```
[ no ] update_source < IP address >
```

example:

```
SR/configure/router/bgp 10/neighbor 101.101.1.2 4> update_source
10.10.2.1
```

applicable systems:

All models.

configure router bgp redistribute

This command provides access to the following next-level commands.

Redistribution causes routes from other protocols to be exported via the current protocol. Routes from the current protocol are always exported, some protocols may provide additional policy features that allow the suppression of protocol routes.

next-level commands
configure router bgp redistribute connected
configure router bgp redistribute ospf
configure router bgp redistribute rip
configure router bgp redistribute static

related commands
configure router bgp aggregate_address
configure router bgp always_compare_med
configure router bgp distance
configure router bgp default_metric
configure router bgp group
configure router bgp neighbor

configure router bgp redistribute connected

This command redistributes interface routes.

parameter	definition
metric	Default metric The range is 0 - 4294967294.
route_map	Name of the route map to use

syntax:

[no] redistribute connected [metric < n >] [route_map < name >]

example:

SR/configure/router/bgp 10> redistribute connected metric 5000

related commands
configure router bgp redistribute ospf
configure router bgp redistribute rip
configure router bgp redistribute static

applicable systems:

All models.

configure router bgp redistribute ospf

This command configures the router to redistribute OSPF routes.

parameter	definition
metric	The default metric The range is 0 - 4294967294.
route_map	Name of the route map to use

syntax:

```
[ no ] redistribute ospf [ metric < n > ] [ route_map < name > ]
```

example:

```
SR/configure/router/bgp 10> redistribute ospf metric 12000
```

related commands
configure router bgp redistribute connected
configure router bgp redistribute rip
configure router bgp redistribute static

applicable systems:

All models.

configure router bgp redistribute rip

This command configures a router to redistribute RIP routes.

parameter	definition
metric	The default metric The range is 0 - 4294967294.
route_map	Name or ID of the route map to use

syntax:

```
[ no ] redistribute rip [ metric < n > ] [ route_map < name > ]
```

example:

```
SR/configure/router/bgp 10> redistribute rip route_map east8
```

related commands
configure router bgp redistribute connected
configure router bgp redistribute ospf
configure router bgp redistribute static

applicable systems:

All models.

configure router bgp redistribute static

This command configures a router to redistribute static routes.

parameter	definition
metric	The default metric The range is 0 - 4294967294.
route_map	Name of the route map to use

syntax:

```
[ no ] redistribute static [ metric < n > ] [ route_map < name > ]
```

example:

```
SR/configure/router/bgp 10> redistribute static metric 25
```

related commands
configure router bgp redistribute connected
configure router bgp redistribute ospf

related commands
configure router bgp redistribute rip

applicable systems:

All models.

Debug ip BGP

Enables or disables BGP4 debug commands

syntax:

```
debug ip bgp
```

example:

```
SR/debug/ip > bgp
```

related commands
debug ip bgp
debug ip bgp all
debug ip bgp events
debug ip bgp neighbor
debug ip bgp packet
debug ip bgp packet all
debug ip bgp events
debug ip bgp neighbor
debug ip bgp packet
debug ip bgp packet all
debug ip bgp packet keepalive
debug ip bgp packet open
debug ip bgp packet update
debug ip bgp policy
debug ip bgp routes
debug ip bgp state
debug ip bgp tasks

```
debug ip bgp timers
```

applicable systems

all models

Chapter 8: BGP4 SHOW COMMANDS

Use BGP show commands to display all configured BGP information.

show ip bgp

This command accesses the following next-level show commands.

next-level commands
show ip bgp aggregate_address
show ip bgp community
show ip bgp groups
show ip bgp neighbors
show ip bgp paths
show ip bgp regexp
show ip bgp summary
show ip bgp table

show ip bgp aggregate_address

This command displays a list of configured aggregate addresses.

parameter	definition
address	Aggregate address Enter an IP address.
mask	Aggregate mask Enter a subnet mask.

syntax:

```
show ip bgp aggregate_address [ address < IP address >[ mask < subnet
mask >]]
```

example:

```
SR> show ip bgp aggregate_address address 100.12.23.0 mask
255.255.255.0
```

applicable systems:

All models:

show ip bgp community

This command displays routes that match BGP communities.

parameter	definition
number	Community number (enter a list of unsigned numbers) The maximum list size is 10. The range is 1 - 4294967295
aa_nn	Community number in aa_nn format Enter a list of strings separated by spaces. The maximum list size is 10 numbers.
match_local_as	
local_as	Do not send outside local AS (well-known community)
match_no_advertise	
no_advertise	Do not advertise to any peer (well-known community)
match_no_export	
no_export	Do not export to next AS (well-known community)
match_exact_match	
exact_match	Exact match of the communities

syntax:

```
show ip bgp community [ number < n > ] [ aa_nn < n > ] [ match_local_as
< local_as > ][ match_no_advertise < no_advertise > ]
[ match_no_export < no_export > ] [ match_exact_match < exact_match
> ]
```

example:

```
SR> show ip bgp community aa_nn 0:999
```

Table 9: Status and Origin Codes

Status codes	
* (valid)	The table entry is valid.
> (best)	The table entry is the best entry to use for that network.
i (internal)	The table entry was learned via an internal BGP session.
Origin codes	
i (IGP)	Internal BGP
e (EGP)	External BGP
? (incomplete)	Protocol of unknown origin. Typically redistributed into BGP from an IGP.

applicable systems:

All models:

show ip bgp groups

This command provides information about BGP groups.

syntax:

```
show ip bgp groups [ < name > ]
```

example:

```
SR> show ip bgp groups north
```

screen display example

> show ip bgp groups

```
BGP group is AvayaBgpExternal
group type is External, total peers 1, established
peers 0, members:
101.202.32.2
options set :
None
```

applicable systems:

All models:

show ip bgp neighbors

This command displays detailed information and status on all BGP neighbors, including:

- peer group and AS affiliations
- configured and negotiated timers
- minimum times between advertisements
- receive and transmit updates
- BGP state status
- TCP connection (active or inactive)

parameter	definition
group	Neighbors belonging to a group Enter a name or word.
address	Neighbor to display information about Enter an IP address.
routes	
advertised	Display the routes advertised to a BGP neighbor.
received	Display the routes received from a neighbor.

syntax:

show ip bgp neighbors [group < name >] [address < IP address>] [routes < advertised | received >]

example:

```
SR> show ip bgp neighbors
```

screen display example

Table 10: Status and Origin Codes

Status codes	
* (valid)	The table entry is valid.
> (best)	The table entry is the best entry to use for that network.
i (internal)	The table entry was learned via an internal BGP session.
Origin codes	
i (IGP)	Internal BGP

e (EGP)	External BGP
? (incomplete)	Protocol of unknown origin.

Table 11: Other BGP show Descriptions

BGP neighbor	IP address of the BGP neighbor
peer group	Displays the name of the peer group.
remote AS	The remote AS number of the neighbor
local AS	The local AS number of the neighbor
link	Identifies the link as internal or external.
BGP version	Identifies the BGP version
local router ID	BGP identifier of the local router
remote router ID	BGP identifier of the reomte router
current state	Current BGP protocol state
last state	Previous BGP protocol state
last event	Previous BGP protocol event
configured hold time	Configured BGP hold time
keepalive interval	Configured BGP keepalive interval
minimum time	Minimum time between advertisements
received	
messages	Number of received BGP messages
notifications	Number of received BGP notifications
updates	Number of received BGP updates
sent	
messages	Number of sent BGP messages
notifications	Number of sent BGP notivications
updates	Number of sent BGP updates
Maximum prefixes	The maximum number of prefixes that can be received from this neighbor.

applicable systems:

All models:

show ip bgp paths

This command shows all BGP paths in the database.

syntax:

```
show ip bgp paths
```

example:

```
SR> show ip bgp paths
```

screen display example

```
> show ip bgp paths
Hash Refcount Path
32   2        ?
96   1        i
/configure>
```

Table 12: Interpreting BGP Paths

term	
hash	An area where path IP addresses are stored
refcount	The number of routes using a specific path
path	The AS path and origin for that route.

Table 13: Status and Origin Codes

Origin codes	
i (IGP)	Internal BGP
e (EGP)	External BGP
? (incomplete)	Protocol of unknown origin.

applicable systems:

All models:

show ip bgp regexp

This command displays routes matching the regular expression.

parameter	definition
reg_exp	A regular expression to match the BGP AS paths. Strings must be enclosed by quotation marks.

syntax:

```
show ip bgp regexp reg_exp < "string" >
```

example:

```
SR> show ip bgp regexp ".* 600 .*"
```

applicable systems:

All models:

show ip bgp summary

This command shows the BGP router's identifying number, local AS number, and connected neighbors. Neighbor information includes BGP version (v), AS number, messages received and transmitted, and operating status.

syntax:

```
show ip bgp summary
```

example:

```
SR> show ip bgp summary
```

screen display example

```
> show ip bgp summary
BGP router identifier 10.1.1.0, local AS member 200
Neighbor        V    AS    MsgRcvd   MsgSent   State
192.168.123.1    4    400      0         0      Active
172.10.16.1      4    200      59        59     Established
>
```

Table 14: Header Definitions

BGP router identifier	The local router ID, IP address
local AS number	The local AS number
V	BGP version spoken by a specific neighbor
AS	Autonomous system
msgRcvd	BGP messages received from a specific neighbor
msgSent	BGP messages sent by a specific neighbor
state	The state of all BGP sessions.

applicable systems:

All models:

show ip bgp table

This command shows entries in the BGP route table.

syntax:

```
show ip bgp table
```

example:

```
SR> show ip bgp table
```

Table 15: Status and Origin Codes

Status codes	
* (valid)	The table entry is valid.
i (internal)	The table entry was learned via an internal BGP session.
Origin codes	
i (IGP)	Internal BGP
e (EGP)	External BGP
? (incomplete)	Protocol of unknown origin.

applicable systems:

All models:

show policy

This command provides access to the following next-level policy display commands:

next-level commands
show policy as_path
show policy community_list
show policy ip_access_list
show policy route_map

show policy as_path

This command displays the AS path access lists.

parameter	definition
access_list	The access list number. The range is 1 - 199.

syntax:

```
show policy as_path [ access_list < n > ]
```

example:

```
SR> show policy as_path
```

screen display example

```
> show policy as_path
AS path access list 1
permit .* 699 .*
permit .* 500
deny 40 .*
AS path access list 2
deny 60.*
>
```

related commands
show policy community_list
show policy ip_access_list

related commands
show policy route_map

applicable systems:

All models:

show policy community_list

This command shows configured community lists.

parameter	definition
community	The community list number. The range is 1 - 199.

syntax:

```
show policy community_list [ community < n > ]
```

example:

```
SR> show policy community_list
```

related commands
show policy as_path
show policy ip_access_list
show policy route_map

applicable systems:

All models:

show policy ip_access_list

This command show routes that comply with specific IP access rules.

parameter	definition
number	IP access list number The range is 1 - 99.

syntax:

```
show policy ip_access_list [ number < n > ]
```

example:

```
SR/show> policy ip_access_list
```

screen display example

related commands
show policy as_path
show policy community_list
show policy route_map

applicable systems:

All models:

show policy route_map

This command shows route map information.

parameter	definition
name	The name of the route map.

syntax:

```
show policy route_map [ < name > ]
```

example:

```
SR> show policy route_map
```

screen display example

```
> show policy route_map
route-map Block100, deny, sequence 1
Batch clauses:
as_path (as-path filter): 99
Set clauses:
origin bgp
>
```

related commands
show policy as_path

related commands
show policy community_list
show policy ip_access_list

applicable systems:

All models:

Chapter 9: OSPF CONFIGURE COMMANDS

Use OSPF configure commands to configure all OSPF routing parameters.

✻ **Note:**

See the command configure interface loopback in the SR1000 *Command Reference Guide* for important information about loopback interfaces.

When configuring OSPF, keep the following in mind:

- When you enable OSPF on bundles, make sure that both ends of the bundle are either "numbered" or "unnumbered." If there is a mismatch, even though the adjacency will come up, route reachability issues may develop.
- When the IP address is specified for a bundle and you later want to change the network type on that bundle to "broadcast," you must also specify the type parameter for the bundle IP address.

  To do this, you must delete the bundle's assigned IP address and reassign the IP address with the type broadcast parameter. For example:

  ```
  SR/configure/interface/bundle wan1> no ip address 2.2.2.2 24
  ```

  ```
  SR/configure/interface/bundle wan1> ip address 2.2.2.2 24 type broadcast
  ```
- Adjacencies will not form if hello_interval, dead_interval, authentication or area_type mismatches are present

configure router ospf

This command configures a router for OSPF routing.

syntax:

```
router ospf
```

example:

```
SR/configure> router ospf
```

next-level commands
configure router ospf 1583Compatability
configure router ospf area

next-level commands
configure router ospf distance
configure router ospf interface
configure router ospf redistribute
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf 1583Compatibility

This command establishes the route summary calculation method to be compatible with RFC 1583. The RFC compatibility of all routers in an OSPF domain should be configured the same.

The default is 1583Compatibility disabled.

syntax:

```
1583Compatibility
```

example:

```
SR/configure/router/ospf> 1583 Compatibility
```

related commands
configure router ospf area
configure router ospf distance
configure router ospf interface
configure router ospf redistribute
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf area

This command configures an OSPF area.

parameter	definition
area_id	OSPF area id Enter either a decimal number or an IP address.

syntax:

```
area < area_id >
```

example:

```
SR/configure/router/ospf> area 0
```

next-level commands
configure router ospf area area_type
configure router ospf area default_cost
configure router ospf area range
configure router ospf area virtual_link

related commands
configure router ospf 1583Compatibility
configure router ospf distance
configure router ospf interface
configure router ospf redistribute
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf area area_type

This command accesses the following next-level commands for configuring an area type.

next-level commands
configure router ospf area area_type normal
configure router ospf area area_type nssa
configure router ospf area area_type stub

related commands
configure router ospf area default_cost
configure router ospf area range
configure router ospf area virtual_link

applicable systems:

All models.

configure router ospf area area_type normal

This command specifies an area area type as normal.

syntax:

```
area_type normal
```

example:

```
SR/configure/router/ospf/area 0> area_type normal
```

related commands
configure router ospf area area_type nssa
configure router ospf area area_type stub

applicable systems:

All models.

configure router ospf area area_type nssa

This command specifies an area type as (nssa) not-so-stubby area.

syntax:

```
area_type nssa
```

example:

```
SR/configure/router/ospf/area 1> area_type nssa
```

next-level commands
configure router ospf area area_type nssa no_summary

related commands
configure router ospf area area_type normal
configure router ospf area area_type stub

applicable systems:

All models.

configure router ospf area area_type nssa no_summary

This command prevents an nssa area boundary router from sending summary link advertisements into an nssa area.

syntax:

```
no_summary
```

example:

```
SR/configure/router/ospf/area 1/area_type/nssa> no_summary
```

applicable systems:

All models.

configure router ospf area area_type stub

This command configures an area as a stub area.

Stub areas are not flooded with AS external advertisements. Stub areas reduce the amount of memory required on stub area routers.

syntax:

```
[ no ] area_type stub
```

example:

SR/configure/router/ospf/area 1> area_type stub

next-level commands
configure router ospf area area_type stub no_summary

related commands
configure router ospf area area_type normal
configure router ospf area area_type nssa

applicable systems:

All models.

configure router ospf area area_type stub no_summary

This command prevents an area boundary router from sending summary link advertisements into the stub area.

syntax:

```
no_summary
```

example:

```
SR/configure/router/ospf/area 1/area_type/stub> no_summary
```

applicable systems:

All models.

configure router ospf area default_cost

This command specifies a cost for the default summary route sent into a stub area.

parameter	definition
default_cost	Enter a number. The range is 0 - 16777215; the default is 1.

syntax:

```
default_cost < n >
```

example:

```
SR/configure/router/ospf/area 1> default_cost 10
```

related commands
configure router ospf area area_type
configure router ospf area range
configure router ospf area virtual_link

applicable systems:

All models.

configure router ospf area range

This command summarizes routes at the area boundaries, producing a single route that is advertised by area border routers.

parameter	definition
networknumber	IP address
mask	netmask
advertise_enum	
advertise	Advertise this range.
not_advertise	Do not advertise this range.

syntax:

```
[ no ] range networknumber < IP address > mask < netmask >
[ advertise_enum < advertise | not_advertise > ]
```

example:

```
SR/configure/router/ospf/area 0> range 100.1.0.0 255.255.0.0
advertise
```

related commands
configure router ospf area_type
configure router ospf area default_cost
configure router ospf area area virtual_link

applicable systems:

All models.

configure router ospf area virtual_link

This command defines an OSPF virtual link for an area.

Establishes a virtual connection to the backbone for an area border router that is not physically connected to the backbone. A virtual link requires that each virtual link neighbor must include the transit area ID and the virtual link neighbor's router ID.

parameter	definition
virtual_link	IP address for the virtual link. Enter an IP address.

syntax:

```
[ no ] virtual_link < IP address >
```

example:

```
SR/configure/router/ospf/area 1> virtual_link 100.10.1.5
```

next-level commands
configure router ospf area virtual_link authentication
configure router ospf area virtual_link dead_interval
configure router ospf area virtual_link hello_interval
configure router ospf area virtual_link retransmit_interval
configure router ospf area virtual_link transmit_delay

applicable systems:

All models.

configure router ospf area virtual_link authentication

This command configures authentication for an area virtual link.

Authentication guarantees that only trusted routers send and receive traffic within an area. Each interface must use the same type of authentication.

parameter	definition
authentication type	
simple	Uses a text password that is imbedded in the packet.
md5	Creates an encoded checksum that is imbedded in the packet.
md5_cisco	Cisco compatible MD5 authentication
line	A 16-character (maximum) password string beginning with an alpha character.

syntax:

```
authentication < none | simple | md5 | md5_cisco > < line >
```

example:

```
SR/configure/router/ospf/area 1/virtual_link 100.10.1.5>
authentication simple Avaya
```

related commands
configure router ospf area virtual_link dead_interval
configure router ospf area virtual_link hello_interval
configure router ospf area virtual_link retransmit_interval
configure router ospf area virtual_link transmit_delay

Note:

Configuring OSPF MD5 Authentication between Secure Router 100x/3120 and the Avaya Ethernet Routing Switch 8600 : In the Secure Router 100x/3120 there are two options for MD5 authentication. The choices are "md5" and "md5_cisco". The differences have to do with the way the hash is generated. When interoperating with the Avaya ERS8600 a user must choose "md5_cisco". The "md5_cisco" option is not for SR interoperation with only Cisco routers. It is considered a more industry standard implementation for use with third party routers. The "md5" option can be used for MD5 authentication between SR100x/3120 units.

applicable systems:

All models.

configure router ospf area virtual_link dead_interval

This command sets the time, in seconds that an OSPF neighbor will wait for a hello packet.

Once the user-defined time expires, the interface assumes that the neighbor is down. The value entered should be approximately four times the value of the hello_interval.

parameter	definition
dead_interval	The time in seconds. The value configured must be the same for all routers and servers in the same network. The range is 1 - 65535; the default value is 40. The recommended value to configure is four times the value configured for the hello interval.

syntax:

```
[ no ] dead_interval < n >
```

example:

```
SR/configure/router/opsf/area 1/virtual_link 100.10.1.5>
dead_interval 10
```

related commands
configure router ospf area virtual_link authentication
configure router ospf area virtual_link hello_interval
configure router ospf area virtual_link retransmit_interval
configure router ospf area virtual_link transmit_delay

applicable systems:

All models.

configure router ospf area virtual_link hello_interval

This command configures the time interval between transmission of hello packets.

parameter	definition
hello_interval	The time in seconds. The value configured must be the same for all routers and servers in the same network. The range is 1 - 65535: the default is 10 seconds.

syntax:

```
[ no ] hello_interval < n >
```

example:

```
SR/configure/router/ospf/area 1/virtual_link 100.10.1.5>
hello_interval 10
```

related commands
configure router ospf area virtual_link authentication
configure router ospf area virtual_link dead_interval
configure router ospf area virtual_link retransmit_interval
configure router ospf area virtual_link transmit_delay

applicable systems:

All models.

configure router ospf area virtual_link retransmit_interval

This command configures the time between link state advertisement retransmissions on an interface.

parameter	definition
retransmit_interval	The time in seconds. The configured value must be greater than the expected round-trip delay. The range is 1 - 65535; the default is 5.

syntax:

```
[ no ] retransmit_interval < n >
```

example:

```
SR/configure/router/ospf/area 1/virtual_link 100.10.1.5>
retransmit_interval 5
```

related commands
configure router ospf area virtual_link authentication
configure router ospf area virtual_link dead_interval
configure router ospf area virtual_link hello_interval
configure router ospf area virtual_link transmit_delay

applicable systems:

All models.

configure router ospf area virtual_link transmit_delay

This command configures the estimated time to transmit a link state update packet on an interface.

parameter	definition
transmit_delay	The time in seconds. Link state advertisements in the update packet are aged by this amount prior to transmission. The range is 1 - 65535; the default is 1. The value must be greater than zero.

syntax:

```
[ no ] transmit_delay < n >
```

example:

```
SR/configure/router/ospf/area 1/virtual_link 100.10.1.5>
transmit_delay 1
```

related commands
configure router ospf area virtual_link authentication
configure router ospf area virtual_link dead_interval
configure router ospf area virtual_link hello_interval
configure router ospf area virtual_link retransmit_interval

applicable systems:

All models.

configure router ospf distance

This command accesses the following next-level commands to configure OSPF administrative distances for routes.

next-level commands
configure router ospf distance ospf

related commands
configure router ospf 1583Compatability
configure router ospf area
configure router ospf interface
configure router ospf redistribute
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf distance ospf

This command accesses next-level commands that configure OSPF administrative distances based on route type.

next-level commands
configure router ospf distance ospf external
configure router ospf distance ospf non_external

applicable systems:

All models.

configure router ospf distance ospf external

This command configures the distance parameter for external routes.

parameter	definition
external	Type-5 and type-7 external routes The range is 1 - 255; the default is 150.

syntax:

```
[ no ] distance ospf external < n >
```

example:

```
SR/configure/router/ospf> distance ospf external 25
```

Table 16: Default Route Preference (Administrative Distance) Values

How Route is Learned	Default Preference	Command to Modify Default Preference
Directly connected network	0	Not configurable.
Static	1	Not configurable.
OSPF non-external route	10	configure router ospf distance ospf non_external
RIP	100	configure router rip distance
Generated or aggregate	130	Applicable to BGP only, and is not configurable.
OSPF AS external routes	150	configure router ospf distance ospf external
BGP	170	configure router bgp distance

related commands
configure router ospf distance ospf non_external

applicable systems:

All models.

configure router ospf distance ospf non_external

This command configures the distance parameter for inter- and intra-area routes.

parameter	definition
non_external	Inter-area and intra-area routes The range is 1 - 255; the default is 10.

syntax:

```
[ no ] distance ospf non_external < n >
```

example:

```
SR/configure/router/ospf> distance ospf non_external 25
```

Table 17: Default Route Preference (Administrative Distance) Values

How Route is Learned	Default Preference	Command to Modify Default Preference
Directly connected network	0	Not configurable.
Static	1	Not configurable.
OSPF non-external route	10	configure router ospf distance ospf non_external
RIP	100	configure router rip distance
Generated or aggregate	130	Applicable to BGP only, and is not configurable.
OSPF AS external routes	150	configure router ospf distance ospf external
BGP	170	configure router bgp distance

related commands
configure router ospf distance ospf external

applicable systems:

All models.

configure router ospf interface

This command configures an interface for OSPF routing.

parameter	definition
name	Enter an interface name, such as ethernet0, ethernet1, or a bundle name.
dlci	Data link connection identifier of the pvc (for frame relay use). The range is 16 - 1022; there is no default.
area_id	OSPF area ID Enter either a decimal number or an IP address.

syntax:

```
[ no ] interface < name > [ dlci < n > ] [ < area _id > ]
```

Note:

When the "ospf" interface is created for the first time, area id must be specified. Thereafter, it is optional.

example:

```
SR/configure/router/ospf> interface Toronto 5
```

next-level commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

related commands
configure router ospf 1583Compatibility
configure router ospf area
configure router ospf distance
configure router ospf redistribute
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf interface authentication

This command configures the authentication type on an interface.

parameter	definition
authentication type	
simple	Simple password authentication
md5	MD5 authentication
md5_cisco	Cisco compatible md5 authentication
line	A 16-character (maximum) password string beginning with an alpha character.

syntax:

```
[ no ] authentication < type > < line >
```

example:

```
SR/configure/router/ospf/interface toBoston> authentication md5
Avaya
```

related commands
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface cost

This command configures the OSPF metric cost for a specific interface.

parameter	definition
cost	Metric cost of sending packets on a particular OSPF interface.

parameter	definition
	The range is 1 - 65535; the default is computed based on the interface bandwidth.

syntax:

```
[ no ] cost < n >
```

example:

```
SR/configure/router/ospf/interface toBoston > cost 10
```

related commands
configure router ospf interface authentication
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface dead_interval

This command sets the time, in seconds, that an OSPF neighbor will wait for a hello packet.

Once the user-defined time expires, the interface assumes that the neighbor is down. The value entered should be approximately four times the value of the hello_interval.

parameter	definition
dead_interval	Time, in seconds The range is 1- 65535; the default is 40.

syntax:

```
[ no ] dead_interval < n >
```

example:

```
SR/configure/router/ospf/interface> dead_interval 50
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface hello_interval

This command sets the time interval, in seconds, between the hello packets that are sent on the interface.

parameter	definition
hello_interval	Time in seconds The default is 10; the range is 1 - 65535.

syntax:

```
[ no ] hello_interval < n >
```

example:

```
SR/configure/router/ospf/interface toBoston> hello_interval 30
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval

related commands
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface neighbor

This command sets up an OSPF neighbor router for an interface that is used on a non-broadcast network.

parameter	definition
ip address	The IP address of the neighbor router
priority	Sets the router priority for a non-broadcast neighbor. The range is 0 - 255; the default is 1.

syntax:

```
[ no ] neighbor < IP address > [ priority < n > ]
```

example:

```
SR/configure/router/ospf/interface toBoston> neighbor 100.22.12.2 7
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority

related commands
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface network

This command configures the OSPF network type on an interface.

interface type	network type default
PPP/HDLC	point-to-point
Ethernet	broadcast
Frame Relay	point-to-point

parameter	definition
network type	
broadcast	Configures network type to broadcast multi-access network
non_broadcast	Configures network type to nonbroadcast multiaccess (NBMA) network
point_to_multipoint	Configures network type to point-to-multipoint network
point_to_point	Configures network type to point-to-point network

syntax:

```
[ no ] network < broadcast | non_broadcast | point_to_multipoint |
point_to_point >
```

Note:

If the interface type is point-to-point, then to change the network type to broadcast, the user must first change the point-to-point interface to broadcast type using the ip address command.

Note:

The "non_broadcast" and "point_to_multipoint" parameters are not supported in this release.

example:

```
SR/configure/router/ospf/interface toBoston> network non_broadcast
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface poll_interval

This command, used for nonbroadcast interfaces only, specifies how often the router sends hello packets from the interface before establishing adjacency with a neighbor.

parameter	definition
poll_interval	The time, in seconds The range is 0 - 2147483647; the default is 120.

syntax:

```
[ no ] poll_interval < n >
```

example:

```
SR/configure/router/ospf/interface toBoston> poll_interval 15
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval

related commands
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface priority
configure router ospf interface retransmit_interval
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface priority

This command configures the priority (which is used in the election of designated routes) to establish the designated router.

parameter	definition
priority	Number that specifies the router priority. This is only used in non point-to-point networks. The range is 0 - 255; the default is 1.

syntax:

```
[ no ] priority < n >
```

example:

```
SR/configure/router/ospf/interface toBoston> priority 5
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface retransmit_interval

related commands
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface retransmit_interval

This command configures the retransmit time for the link state advertisement retransmission for neighbors belonging to the interface.

When a router sends a link state advertisement to its neighbor, it keeps the LSA until it receives an acknowledgment. If an acknowledgment is not received in n seconds, the router will retransmit the LSA.

parameter	definition
seconds	Time in seconds between retransmission. It must be conservatively set, but greater than the expected round trip delay between routers on the attached network. The range is 1- 65535; the default is 5.

syntax:

```
[ no ] retransmit_interval < n >
```

example:

```
SR/configure/router/ospf/interface toBoston> retransmit_interval 60
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface transmit_delay

This command configures the approximate time it takes to transmit a link state advertisement update packet on the interface.

parameter	definition
seconds	Time in seconds. Usage of this command is most appropriate for low speed links. The range is 1- 65535; the default is 1.

syntax:

```
[ no ] transmit_delay < n >
```

example:

```
SR/router/ospf/interface toBoston> transmit_delay 3
```

related commands
configure router ospf interface authentication
configure router ospf interface cost
configure router ospf interface dead_interval
configure router ospf interface hello_interval
configure router ospf interface neighbor
configure router ospf interface network
configure router ospf interface poll_interval
configure router ospf interface priority
configure router ospf interface retransmit_interval

applicable systems:

All models.

configure router ospf redistribute

This command accesses next-level commands that are used to redistribute routes from other routers or routing protocols.

syntax:

```
redistribute
```

example:

```
SR/configure/router/ospf> redistribute
```

next-level commands
configure router ospf redistribute bgp
configure router ospf redistribute connected
configure router ospf redistribute rip
configure router ospf redistribute static

related commands
configure router ospf 1583Compatability
configure router ospf area
configure router ospf distance
configure router ospf interface
configure router ospf ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf redistribute bgp

This command redistributes BGP routes.

parameter	definition
as_number	Autonomous system number The range is 1 - 65535.

parameter	definition
metric	OSPF default metric The range is 0 - 16777214; the default is 100.
metric_type	Ospf exterior metric type for redistribution The range is 1 - 2; the default is 2.
route_map	Pointer (name or word) to route map entries
tag	32-bit tag value The range is 0 - 2147483647; the default is 0.

Note:

See the Policy commands chapter, specifically configure policy route_map on page 32 for more information about configuring route maps.

syntax:

```
redistribute bgp as_number < n > [ metric < n > ] [ < metric_type < n
> ] [ route_map < name > ] [ tag < n > ]
```

example:

```
SR/configure/router/ospf> redistribute bgp as_number 10
```

related commands
configure router ospf redistribute connected
configure router ospf redistribute rip
configure router ospf redistribute static

applicable systems:

All models.

configure router ospf redistribute connected

This command redistributes connected interface routes.

parameter	definition
metric	OSPF default metric The range is 0 - 16777214; the default is 100.
metric_type	Ospf exterior metric type for redistribution The range is 1 - 2; the default is 2.
route_map	Pointer (name or word) to route map entries

parameter	definition
tag	32-bit tag value The range is 0 - 2147483647; the default is 0.

Note:
See policy commands, specifically configure policy route_map on page 32 for more information about configuring route maps.

syntax:

```
redistribute connected [ metric < n > ] [ < metric_type < n > ]
[ route_map < name > ] [ tag < n > ]
```

example:

```
SR/configure/router/ospf> redistribute connected
```

related commands
configure router ospf redistribute bgp
configure router ospf redistribute rip
configure router ospf redistribute static

applicable systems:

All models.

configure router ospf redistribute rip

This command redistributes RIP routes.

parameter	definition
metric	OSPF default metric The range is 0 - 16777214; the default is 100.
metric_type	Ospf exterior metric type for redistribution The range is 1 - 2; the default is 2.
route_map	Pointer (name or word) to route map entries
tag	32-bit tag value The range is 0 - 2147483647; the default is 0.

Note:

See policy commands, specifically configure policy route_map on page 32 for more information about configuring route maps.

syntax:

```
redistribute rip [ metric < n > ] [ < metric_type < n > ] [ route_map <
name > ] [ tag < n > ]
```

example:

```
SR/configure/router/ospf> redistribute rip
```

related commands
configure router ospf redistribute bgp
configure router ospf redistribute connected
configure router ospf redistribute static

applicable systems:

All models.

configure router ospf redistribute static

This command redistributes static routes.

parameter	definition
metric	OSPF default metric The range is 0 - 16777214; the default is 100.
metric_type	Ospf exterior metric type for redistribution The range is 1 - 2; the default is 2.
route_map	Pointer (name or word) to route map entries
tag	32-bit tag value The range is 0 - 2147483647; the default is 0.

syntax:

```
redistribute static [ metric < n > ] [ < metric_type < n > ]
[ route_map < name > ] [ tag < n > ]
```

example:

```
SR/configure/router/ospf> redistribute static
```

related commands
configure router ospf redistribute bgp
configure router ospf redistribute connected
configure router ospf redistribute static

applicable systems:

All models.

configure router ospf ref_bw

This command calculates OSPF interface cost according to bandwidth usage.

Specifying a large number helps differentiate cost on multiple high bandwidth links.

parameter	definition
reference_bandwidth	Reference bandwidth in Mbps The range is 1 - 4294967.

syntax:

```
ref_bw < n >
```

example:

```
SR/configure/router/ospf> ref_bw 100000
```

related commands
configure router ospf 1583Compatability
configure router ospf area
configure router ospf distance
configure router ospf interface
configure router ospf redistribute
configure router ospf timers

applicable systems:

All models.

configure router ospf timers

This command configures and adjusts ospf spf timers.

parameter	definition
timers	
spf_delay	Delay between receiving a change to the SPF calculation. The range is 1 - 65535; the default is 5.
spf_holdtime	The hold time between consecutive SPF calculations. The range is 1 - 65535; the default is 10.

syntax:

```
timers [ spf_delay < n > | spf_holdtime < n > ]
```

example:

```
SR/configure/router/ospf> timers spf_delay 20
```

related commands
configure router ospf 1583Compatibility
configure router ospf area
configure router ospf distance
configure router ospf interface
configure router ospf redistribute
configure router ospf ref_bw

applicable systems:

All models.

Debug ip ospf

Enables or disables OSPF debug commands

syntax:

```
debug ip ospf
```

example:

```
SR/debug/ip > ospf
```

related commands
debug ip ospf
debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf packet all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_request
debug ip ospf packet ls_update
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable systems

all models

Chapter 10: AS PATH REGULAR EXPRESSIONS

This appendix provides information about how to use and configure regular expressions for use with BGP4 routing protocol commands.

Matching AS Paths

An AS path regular expression is a regular expression with the alphabet used as the set of AS numbers defining a set of AS paths.

Note that according to this definition, AS path regular expressions are implicitly anchored at the beginning and end.

The following examples provide more information:

690	Matches only the specific AS path "690."
. *690 .*	Matches any AS path containing 690.
690 .*	Matches any AS path beginning with 690.
. *690	Matches any AS path ending in 690.

AS Path Regular Expressions (regex)

A regex is a character string containing one of the following:

term	Matches the given term.
regex1 regex2	Matches a path that is a concatenation of two paths, P1 and P2. P1 matches regex1 and P2 matches regex2. Note thatspaces are ignored in general, but should be used between two concatenated ASs to distunguish them.
regex1 |regex2	Matches a path that matches regex1 or regex2.

AS Path Terms

A term is one of the following:

AS	Matches the given number, which is any positive 16-bit number from 0-65535 inclusive. Note that valid AS numbers range from one through 65534 inclusive.
!AS	Matches any AS number except the given one.
AS1 -AS2	Is a range of ASs. It matches all AS numbers between AS1 and AS2 inclusive.
!AS1 - AS2	This matches all numbers except the given one.
.	Matches any number.
null	Matches an empty (0 length) string, e.g., (AS1 empty AS2) is equivalent to (AS1 AS2).
term {m, n}	A term followed by {m, n} (where m and n are both non-negative integers and m <= n) means at least m and at most n repetitions.
term {m}	A term followed by {m} (where m is a positive integer) matches m or more repetitions of term.
term {m,}	A term followed by {m,} (where m is a positive integer) matches m or more repetitions of term
term *	A term followed by * matches zero or more repetitions of term. This is shorthand for {0,}.
term +	A term followed by + matches one or more repetitions of term. This is shorthand for {1,}.
term ?	A term followed by ? matches zero or one repetition of term. This is shorthand for {0,1}.
[as_range_list]	Brackets union the items of an as_range_list. An item of this list can be either an AS or a range. For example, {AS1 AS2 - AS3 AS4] is equivalent to (AS1 | AS2-AS3 | AS4).
(regex)	Parentheses group expressions to make a term out of any regex. An operator, such as * or ?, works on a regular expression enclosed in parentheses as it would any term.

Chapter 11: OSPF SHOW COMMANDS

Use OSPF display/show commands to display all configured OSPF information.

show ip ospf area

This command displays configuration information about an OSPF area.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.

syntax:

```
area [ area_id ]
```

example:

```
SR> show ip ospf area 1
```

screen display example

```
> show ip ospf area_id 1
Area 1
Number of interfaces in this area is 0
Area type is NORM
SPF algorithm executed 1 times
No area ranges defined
>
```

related commands
show ip ospf global
show ip ospf database
show ip ospf interface
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf database

This command provides access to commands that display information about an OSPF database.

syntax:

```
database
```

example:

```
SR> show ip ospf database
```

next-level commands
show ip ospf database all
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

related commands
show ip ospf area
show ip ospf global
show ip ospf interface
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf database all

This command displays information related to the OSPF databases of the router.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
show ip ospf database all [ area_id < n > ] [ advt_rtr < IP address
> ] [ link_id < IP address > ]
```

example:

```
SR> show ip ospf database all
```

screen display example

next-level commands
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database asbr_summary

This command displays information about ASBR summary link states.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
database asbr_summary [ area_id < decimal form or IP address > ]
[ advt_rtr < IP address > ] [ link_id < IP address > ]
```

example:

```
SR> show ip ospf database asbr_summary
```

next-level commands
show ip ospf database all
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database database_summary

This command displays OSPF database summary information.

syntax:

```
database database_summary
```

example:

```
SR> show ip ospf database database_summary
```

screen display example

next-level commands
show ip ospf database all
show ip ospf database asbr_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database external

This command displays information about external LSAs in the OSPF database.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
database external [area_id < decimal form or IP address >] [ advt_rtr
< IP address > ] [ link_id < IP address >]
```

example:

```
SR> show ip ospf database external
```

next-level commands
show ip ospf database all
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database network

This command displays database information about the network LSAs.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
database network [area_id < decimal form or IP address >] [ advt_rtr
< IP address >][ link_id < IP address > ]
```

example:

```
SR> show ip ospf database network
```

next-level commands
show ip ospf database all
show ip ospf database asbr_summary

next-level commands
show ip ospf database database_summary
show ip ospf database external
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database nssa_external

This command shows OSPF database information about nssa external LSAs.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
database nssa_external [area_id < decimal value or IP address > ]
[ advt_rtr < IP address >] [ link_id < IP address > ]
```

example:

```
SR > show ip ospf database nssa_external
```

next-level commands
show ip ospf database all
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network

next-level commands
show ip ospf database router
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database router

This command shows information about router LSAs in the OSPF database.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

```
database router [area_id < decimal form or IP address > ] [ advt_rtr
< IP address > ] [ link_id < IP address > ]
```

example:

```
SR> show ip ospf database router
```

```
> show ip ospf database router
Router LSAs for Area 0
LS age: 1743
LS Options: ( E  )
Link State ID: 10.1.1.1
Advertising Router: 10.1.1.1
LS Sequence>: 0x80000001
LS Checksum: 0x6cdc
Length: 24
Number of Links: 0
>
```

next-level commands
show ip ospf database all

next-level commands
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database self_originate
show ip ospf database summary

applicable systems:

All models.

show ip ospf database self_originate

This command displays OSPF database information about self-originated LSAs in the router.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.

syntax:

```
database self_originate [area_id < n > ]
```

example:

```
SR> show ip ospf database self_originate
```

screen display example

next-level commands
show ip ospf database al
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router

next-level commands
show ip ospf database summary

applicable systems:

All models.

show ip ospf database summary

This command displays information about summary LSAs in the OSPF database.

parameter	definition
area_id	OSPF area ID Enter either a decimal number or an IP address.
advt_rtr	OSPF advertisement router Enter an IP address.
link_id	OSPF link state ID Enter an IP address.

syntax:

database summary [area_id < decimal form or IP address >] [advt_rtr < IP address >] [link_id < IP address >]

example:

SR> show ip ospf database summary

next-level commands
show ip ospf database all
show ip ospf database asbr_summary
show ip ospf database database_summary
show ip ospf database external
show ip ospf database network
show ip ospf database nssa_external
show ip ospf database router
show ip ospf database self_originate

applicable systems:

All models.

show ip ospf global

This command displays global OSPF information.

syntax:

```
global
```

example:

```
SR> show ip ospf global
```

screen display example

related commands
show ip ospf area
show ip ospf database
show ip ospf interface
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf interface

This command provides access to commands that display information about configured OSPF interfaces.

syntax:

```
interface
```

example:

```
SR> show ip ospf interface
```

next-level commands
show ip ospf interface all
show ip ospf interface bundle
show ip ospf interface ethernet

related commands
show ip ospf area
show ip ospf global
show ip ospf database
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf interface all

This command displays configuration information about all configured OSPF interfaces.

syntax:

```
interface all
```

example:

```
SR> show ip ospf interface all
```

related commands
show ip ospf interface bundle
show ip ospf interface ethernet

applicable systems:

All models.

show ip ospf interface bundle

This command displays configuration information about an OSPF bundle.

syntax:

```
interface bundle < name > [ pvc < n > ]
```

example:

```
SR> show ip ospf interface bundle Boise
```

related commands
show ip ospf interface all
show ip ospf interface ethernet

applicable systems:

All models.

show ip ospf interface ethernet

This command displays OSPF configuration information about an Ethernet interface.

syntax:

```
interface ethernet < n >
```

example:

```
SR> show ip ospf interface ethernet 1
```

related commands
show ip ospf interface all
show ip ospf interface bundle

applicable systems:

All models.

show ip ospf neighbor

This command provides access to next-level commands that display configuration information about OSPF neighbors.

syntax:

```
neighbor
```

example:

```
SR> show ip ospf neighbor
```

next-level commands
show ip ospf neighbor detail
show ip ospf neighbor id
show ip ospf neighbor interface
show ip ospf neighbor list

related commands
show ip ospf area
show ip ospf global
show ip ospf database
show ip ospf interface
show ip ospf retransmission_list
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf neighbor detail

This command displays detailed OSPF configuration information about all neighbors.

syntax:

```
neighbor detail
```

example:

```
SR> show ip ospf neighbor detail
```

related commands
show ip ospf neighbor id
show ip ospf neighbor interface
show ip ospf neighbor list

applicable systems:

All models.

show ip ospf neighbor id

This command displays OSPF configuration information about a specific neighbor.

syntax:

```
neighbor id < IP address >
```

example:

```
SR> show ip ospf neighbor id 10.3.1.2
```

related commands
show ip ospf neighbor detail
show ip ospf neighbor interface
show ip ospf neighbor list

applicable systems:

All models.

show ip ospf neighbor interface

This command provides access to commands that display OSPF configuration information about all neighbors in an interface.

syntax:

```
neighbor interface ethernet < n > | bundle < name > [ pvc < n > ]
```

example:

```
SR> show ip ospf neighbor interface ethernet 1
```

next-level commands
show ip ospf neighbor interface bundle
show ip ospf neighbor interface ethernet

applicable systems:

All models.

show ip ospf neighbor interface bundle

This command displays information about an OSPF neighbors on a bundle interface.

syntax:

```
neighbor interface bundle < name > [ pvc < n > ]
```

example:

```
SR> show ip ospf neighbor interface bundle Boise
```

related commands
show ip ospf neighbor interface ethernet

applicable systems:

All models.

show ip ospf neighbor interface ethernet

This command displays configuration information about a neighbor on an Ethernet interface.

syntax:

```
neighbor interface ethernet < n >
```

example:

```
SR> show ip ospf neighbor interface ethernet 1
```

related commands
show ip ospf neighbor interface bundle

applicable systems:

All models.

show ip ospf neighbor list

This command displays a list of neighbors attached to this router.

syntax:

```
neighbor list
```

example:

```
SR> show ip ospf neighbor list
```

related commands
show ip ospf neighbor detail
show ip ospf neighbor id
show ip ospf neighbor interface

applicable systems:

All models.

show ip ospf request_list

This command displays the LSAs in the request list of the specified neighbor.

syntax:

```
request_list < IP address >
```

example:

```
SR> show ip ospf request_list 10.10.10.1
```

related commands
show ip ospf area

related commands
show ip ospf global
show ip ospf database
show ip ospf interface
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf retransmission_list

This command displays the LSAs in the retransmission list of the specified neighbor.

syntax:

```
retransmission_list < IP address >
```

example:

```
SR> show ip ospf retransmission_list 10.10.10.1
```

related commands
show ip ospf area
show ip ospf database
show ip ospf global
show ip ospf interface
show ip ospf neighbor
show ip ospf request_list
show ip ospf virtual_links

applicable systems:

All models.

show ip ospf virtual_links

This command displays information about configured OSPF virtual links.

syntax:

```
virtual_links [ < IP address > ]
```

example:

```
SR> show ip ospf virtual_links
```

related commands
show ip ospf area
show ip ospf global
show ip ospf database
show ip ospf interface
show ip ospf neighbor
show ip ospf retransmission_list
show ip ospf request_list

applicable systems:

All models.

Chapter 12: RIP CONFIGURE COMMANDS

Use RIP configure commands to configure all RIP parameters.

Note:

See the command configure interface loopback in the Secure Router 1000 *Command Reference Guide* for important information about loopback interfaces.

configure router rip

This command enables the Routing Information Protocol (RIP).

syntax:

```
[ no ] router rip
```

example:

```
SR/configure> router rip
```

next-level commands
configure router rip default_metric
configure router rip distance
configure router rip interface
configure router rip mode
configure router rip pacing
configure router rip passive
configure router rip redistribute
configure router rip timers

applicable systems:

All models.

configure router rip default_metric

This command sets the global default metric values for RIP.

parameter	definition
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

```
[ no ] default_metric < n >
```

example:

```
SR/configure/router/rip> default_metric 4
```

This example configures the default metric to 4.

related commands
configure router rip distance
configure router rip interface
configure router rip mode
configure router rip pacing
configure router rip passive
configure router rip redistribute
configure router rip timers

applicable systems:

All models.

configure router rip distance

This command configures the distance value for RIP protocol on a router.

parameter	definition
distance	Distance value (enter a number) The range is 1 - 255; the default is 100.

syntax:

```
distance < n >
```

example:

```
SR/configure/router/rip> distance 25
```

Table 18: Default Route Preference (Administrative Distance) Values

How Route is Learned	Default Preference	Command to Modify Default Preference
Directly connected network	0	Not configurable.
Static	1	Not configurable.
OSPF internal route	10	configure router ospf distance ospf non_external
RIP	100	configure router rip distance
Generated or aggregate	130	Applicable to BGP only, and is not configurable.
OSPF AS non-external route	150	configure router ospf distance ospf external
BGP	170	configure router bgp distance

related commands
configure router rip default_metric
configure router rip interface
configure router rip mode
configure router rip pacing
configure router rip passive
configure router rip redistribute
configure router rip timers

applicable systems:

All models.

configure router rip interface

This command enables RIP for an interface.

The interface is identified by the interface name. Use ethernet0 for Ethernet 0 and ethernet1 for Ethernet 1. WAN interfaces are identified by bundle names. If no other RIP interface command is given, then the interface is configured with default RIP parameters.

parameter	definition
name	ethernet0, ethernet1, or a bundle name
dlci	PVC identifier; enter a number. Use only for an encapsulated fr bundle. The range is 16 - 1022.

syntax:

```
[ no ] configure router rip interface < name > [ dlci < n > ]
```

example:

```
SR/configure/router/rip> interface ethernet0
```

This example configures the Ethernet 0 interface for RIP.

next-level commands
configure router rip interface authentication
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface authentication

This command configures RIP-2 authentication for an interface.

The type of authentication and the key value to be used can be specified, but this is only valid with RIP version 2 (mode 3). When authentication is configured, all subsequent RIP updates contain authentication information. In addition, all subsequent incoming RIP packets on that interface are accepted only if they carry a valid authentication header.

parameter	definition
auth_type	The RIP-2 authentication algorithm.
simple	Use simple password authentication.
md5	Use MD5 authentication.
md5_cisco	Use Cisco MD5 compatibility.
line	The RIP-2 authentication password/key Enter an alphanumeric string of up to a maximum of 16 characters.

syntax:

```
[ no ] authentication auth_type line
```

example:

```
SR/configure/router/rip/interface ethernet1> authentication md5
mymd5keyvalue
```

This example configures RIP interface Ethernet 1 for MD5 authentication.

related commands
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface distribute_list

This command configures the access list to be used to filter either incoming or outgoing routes for this interface.

This command is used in conjunction with the redistribute command.

parameter	definition
access_list	Access list number

parameter	definition
	Enter a number.
direction	Traffic flow direction
in	Inbound
out	Outbound

syntax:

```
[ no ] distribute_list < n > < in | out >
```

example:

```
SR/configure/router/rip/interface ethernet0> distribute_list 2 in
```

This example sets access list >2 to be used for all inbound routes for this interface.

related commands
configure router rip interface authentication
configure router rip interface metric
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface metric

This command configures the metric value for RIP routes for this interface.

parameter	definition
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

```
[ no ] metric < n >
```

example:

```
SR/configure/router/rip/interface ethernet0> metric 3
```

This example configures the RIP routes metric for interface Ethernet 0 to 3.

related commands
configure router rip interface authentication
configure router rip interface distribute_list
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface mode

This command configures RIP mode for the specific interface.

This command is similar to the global RIP mode command, but it is only applicable to the current interface. Use this command to override the global RIP mode settings.

parameter	definition
mode	Enter a mode value.
1	RIP version 1
2	RIP version 2 (default)
3	RIP version 2 (V1 compatible)

syntax:

```
[ no ] mode < n >
```

example:

```
SR/configure/router/rip/interface ethernet0> mode 1
```

This example configures interface Ethernet 0 for RIP version 1.

related commands
configure router rip interface authentication

related commands
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface neighbor
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface neighbor

This command specifies a RIP neighbor for a specific interface.

Use this command multiple times to add multiple neighbors. When neighbors are specified, RIP updates are unicast to those neighbors (and not broadcast or multicast on that segment).

parameter	definition
ip_address	Neighbor IP address

syntax:

```
[ no ] neighbor < ip_address >
```

example:

```
SR/configure/router/rip/interface ethernet0> neighbor 192.168.31.2
```

This example configures IP address 192.168.31.2 as a RIP neighbor of interface Ethernet 0.

related commands
configure router rip interface authentication
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface mode
configure router rip interface passive
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface passive

This command configures RIP mode for a specific interface to passive (listen-only) mode.

Use this command to override a global RIP mode configured for an interface.

syntax:

```
[ no ] passive
```

example:

```
SR/configure/router/rip/interface ethernet1> passive
```

This example configures interface Ethernet 1 to listen-only mode.

related commands
configure router rip interface authentication
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface split_horizon

applicable systems:

All models.

configure router rip interface split_horizon

This command configures the split-horizon mechanism on an interface.

By default, split horizon is enabled for all interfaces for poison-reverse.

parameter	definition
splitval	Split horizon algorithm
none	Disables split horizon.
simple	Enables split horizon.

parameter	definition
poison	Enables poison reverse (default)

syntax:

```
[ no ] split_horizon < none | simple | poison >
```

example:

```
SR/configure/router/rip/interface ethernet0> split_horizon simple
```

This example configures interface Ethernet 0 to do simple split-horizon.

related commands
configure router rip interface authentication
configure router rip interface distribute_list
configure router rip interface metric
configure router rip interface mode
configure router rip interface neighbor
configure router rip interface passive

applicable systems:

All models.

configure router rip mode

This command globally configures RIP mode for all interfaces.

Use this command to override the global mode setting.

parameter	definition
mode	Enter a mode value.
1	RIP version 1
2	RIP version 2 (default)
3	RIP version 2 (V1 compatible)

syntax:

```
[ no ] mode < n >
```

example:

```
SR/configure/router/rip> mode 3
```

related commands
configure router rip default_metric
configure router rip distribute_list
configure router rip interface
configure router rip passive
configure router rip distance
configure router rip redistribute

applicable systems:

All models.

configure router rip pacing

This command enables RIP updates sent from this router to be released to the network in a controlled manner to avoid traffic bottlenecks.

When enabled, RIP updates from this router will be sent in several small intervals instead on one burst. This is useful when the number of routes to be sent is large (more than 1000).

syntax:

```
[ no ] pacing
```

example:

```
SR/configure/router/rip> pacing
```

related commands
configure router rip default_metric
configure router rip distance
configure router rip interface
configure router rip mode
configure router rip passive
configure router rip redistribute
configure router rip timers

applicable systems:

All models.

configure router rip passive

This command configures RIP passive (listen only) mode.

All configured interfaces will only listen to RIP (version 1 and 2) updates, but will not send any updates. You can override the mode on a specific interface by configuring RIP mode for that specific interface.

syntax:

```
[ no ] passive
```

example:

```
SR/configure/router/rip> passive
```

This example configures all RIP interfaces to listen-only mode.

related commands
configure router rip default_metric
configure router rip distance
configure router rip interface
configure router rip default mode
configure router rip pacing
configure router rip redistribute
configure router rip timers

applicable systems:

All models.

configure router rip redistribute

This command accesses the following next-level commands that configure the system to use RIP updates to redistribute routes learned from other routing protocols.

next-level commands
configure router rip redistribute bgp

next-level commands
configure router rip redistribute connected
configure router rip redistribute ospf
configure router rip redistribute static

applicable systems:

All models.

configure router rip redistribute bgp

This command configures RIP to redistribute bgp routes.

parameter	definition
as_number	Autonomous system number The range is 1 - 65535.
metric	Default metric The range is 1 - 16; the default is 1.
route_map	Word pointer to route map entries

syntax:

```
redistribute bgp as_number [ metric ] [route_map]
```

example:

```
SR/configure/router/rip> redistribute bgp 1
```

related commands
configure router rip redistribute connected
configure router rip redistribute ospf
configure router rip redistribute static

applicable systems:

All models.

configure router rip redistribute connected

This command configures RIP to redistribute connected routes.

parameter	definition
metric	Default metric The range is 1 - 16; the default is 1.
route_map	Word pointer to route map entries

syntax:

```
[ no ] redistribute connected [ metric ] [route_map]
```

example:

```
SR/configure/router/rip> redistribute connected
```

This example configures RIP to redistribute connected routes.

related commands
configure router rip redistribute bgp
configure router rip redistribute ospf
configure router rip redistribute static

applicable systems:

All models.

configure router rip redistribute ospf

This command configures RIP to redistribute OSPF routes.

parameter	definition
metric	Default metric The range is 1 - 16; the default is 1.
route_map	Word pointer to route map entries

```
[ no ] redistribute ospf [ metric ] [route_map]
```

example:

```
SR/configure/router/rip> redistribute ospf
```

related commands
configure router rip redistribute bgp
configure router rip redistribute connected
configure router rip redistribute static

applicable systems:

All models.

configure router rip redistribute static

This command configures RIP to redistribute static routes.

parameter	definition
metric	Default metric The range is 1 - 16; the default is 1.
route_map	Word pointer to route map entries

syntax:

```
[ no ] redistribute static [ metric ] [route_map]
```

example:

```
SR/configure/router/rip> redistribute static
```

This example configures RIP to redistribute static routes.

related commands
configure router rip redistribute bgp
configure router rip redistribute connected
configure router rip redistribute ospf

applicable systems:

All models.

configure router rip timers

This command accesses the following next-level commands that configure the global RIP timers.

next-level commands
configure router rip timers flush
configure router rip timers holddown
configure router rip timers update

applicable systems:

All models.

configure router rip timers flush

This command configures the global RIP flush timer.

This is the time interval in seconds that must pass before the route is removed from the routing table. This value should be configured to be greater than the configured holddown time value.

parameter	definition
time	Flush timer value in seconds The range is 1 - 65535; the default is 180.

syntax:

```
[ no ] flush time < n >
```

example:

```
SR/configure/router/rip/timers> flush 300
```

This example configures the global RIP flush timer to 300 seconds.

related commands
configure router rip timers holddown
configure router rip timers update

applicable systems:

All models.

configure router rip timers holddown

This command configures the global RIP hold down timers.

Hold down time is the interval in seconds during which routing information regarding better routes is suppressed. This should be configured to be at least twice the value of the update timers.

parameter	definition
time	Holddown timer value in seconds The range is 1- 65535; the default is 180.

syntax:

```
[ no ] holddown time < n >
```

example:

```
SR/configure/router/rip/timers> holddown 200
```

This example configures the global RIP hold down timers to suppress information about routes for 200 seconds.

related commands
configure router rip timers flush
configure router rip timers update

applicable systems:

All models.

configure router rip timers update

This command configures the global RIP update timer.

This timer specifies the interval in seconds for sending periodic RIP updates.

parameter	definition
time	Update timer in seconds The range is 1 - 65535; the default is 120.

syntax:

```
[ no ] update time < n >
```

example:

```
SR/configure/router/rip/timers> update 45
```

This example globally configures RIP updates to occur every 45 seconds.

related commands
configure router rip timers flush
configure router rip timers holddown

applicable systems:

All models.

Debug ip RIP

Enables or disables RIP debug commands

syntax:

```
debug ip rip
```

example:

```
SR/debug/ip > rip
```

related commands
debug ip rip
debug ip rip all
debug ip rip detail
debug ip rip flood
debug ip rip packet
debug ip rip state

applicable systems

all models

Chapter 13: RIP SHOW COMMANDS

Use RIP display/show commands to display all configured RIP information.

show ip rip

This command accesses the following next-level commands that display more specific information.

next-level commands
show ip rip global
show ip rip interface
show ip rip statistics

applicable systems

All models.

show ip rip global

This command displays global configured information about mode, distance, default metric, and timers for RIP.

syntax:

```
show ip rip global
```

example:

```
SR> show ip rip global
```

screen display example

```
> show ip rip global
Router RIP is enabled
Mode: RIP 2
Distance: 100
```

```
Default Metric: 1
Timers:
Update: 30 seconds
Holddown: 120 seconds
Flush: 180 seconds
>
```

related commands
show ip rip interface
show ip rip routes
show ip rip statistics

applicable systems:

All models.

show ip rip interface

This command accesses the following next-level commands that display configuration information about mode, metric, authentication, split horizon, and routers for the RIP interface.

next-level commands
show ip rip interface all
show ip rip interface bundle
show ip rip interface ethernet
show ip rip interface statistics

applicable systems:

All models.

show ip rip interface all

This command displays information about all configured RIP interfaces.

syntax:

```
show ip rip interface all
```

example:

```
SR > show ip rip interface all
```

screen display example

```
> show ip rip interface all
RIP is configured for interface <ethernet0>
Mode: RIP 2
Metric: 5
Authentication: None
Split Horizon: Poison
Routers : None
Interface state:  Broadcast Multicast Active
>
```

related commands
show ip rip interface bundle
show ip rip interface ethernet
show ip rip interface statistics

applicable systems:

All models.

show ip rip interface bundle

This command displays RIP information for a configured bundle.

parameter	definition
bundle_name	The name of the desired bundle. Enter a string of up to a maximum of 8 characters.
pvc	PVC identifier Used only for an encapsulated fr bundle. The range is 16 - 1022.

syntax:

```
show ip rip interface bundle < name >
```

example:

```
SR> show ip rip interface bundle Dallas
```

related commands
show ip rip interface all

related commands
show ip rip interface ethernet
show ip rip interface statistics

applicable systems:

All models.

show ip rip interface ethernet

This command displays RIP information about the Ethernet interface.

syntax:

```
show ip rip interface ethernet < 0 | 1 >
```

example:

```
SR> show ip rip interface ethernet0
```

screen display example

```
> show ip rip interface ethernet 0
RIP is configured for interface <ethernet0>
Mode: RIP 2
Metric: 5
Authentication: None
Split Horizon: Poison
Routers : None
Interface state:  Broadcast Multicast Active
>
```

related commands
show ip rip interface all
show ip rip interface bundle
show ip rip interface statistics

applicable systems:

All models.

show ip rip interface statistics

This command displays global RIP interface statistics, such as the number of pad packets received, the number or bad routes received, and the number of triggered updates sent.

syntax:

```
show ip rip interface statistics
```

example:

```
SR> show ip rip interface statistics
```

screen display example

```
> show ip rip interface statistics
RIP Interface Statistics:
=========================
Interface: <ethernet0>
Number of bad packets received   : <0>
Number of bad routes received    : <0>
Number of Triggered updates sent : <0>
>
```

related commands
show ip rip interface all
show ip rip interface bundle
show ip rip interface ethernet

applicable systems:

All models.

show ip rip statistics

This command shows global RIP statistics, such as route changes and queries.

syntax:

```
show ip rip statistics
```

example:

```
SR> show ip rip statistics
```

screen display example

```
show ip rip statistics
RIP Global Statistics:
===============================
Number of Global Route Changes : <0>
Number of Global Queries       : <0>
>
```

related commands
show ip rip global
show ip rip interface
show ip rip routes

applicable systems:

All models.

Chapter 14: MULTICASTING

Multicasting Overview

Traditional multicast routing mechanisms such as Distance Vector Multicast Routing Protocol (DVMRP) and Multicast Open Shortest Path First (MOSPF) were intended for use within regions where groups are densely populated or bandwidth is universally plentiful. When groups, and senders to these groups, are distributed sparsely across a wide area, these "dense mode" schemes do not perform efficiently.

Protocol Independent Multicast (PIM)

Protocol Independent Multicast (PIM) protocols route multicast packets to multicast groups. PIM is protocol independent because it can leverage whichever unicast routing protocol is used to populate unicast routing table. There are two modes of PIM protocol – Dense mode (DM) and Sparse mode (SM). Avaya supports SM only.

PIM-DM floods multicast traffic throughout the network initially and then generates prune messages as required. PIM-SM attempts to send multicast data only to networks which have active receivers. This is achieved by having a common Rendezvous Point (RP) known to the senders and receivers and by forming shared trees from the RP to the receivers.

PIM-SM is described in RFC 2362.

PIM Commands

The general PIM commands supported in this release are:

Global parameters	
Enable PIM	SR/configure/ip> pim
Configure PIM mode	SR/configure/ip/pim> mode [sparse | dense]
Configure Assert Holdtime	SR/configure/ip/pim> assert-holdtime <time>
Configure Hello Interval	SR/configure/ip/pim> hello-interval <time>
Configure Hello Holdtime	SR/configure/ip/pim> hello-holdtime <time>

Configure Hello priority	SR/configure/ip/pim> hello-priority <value>
Configure Join/Prune Holdtime	SR/configure/ip/pim> join-prune-holdtime <time>
Configure Join /Prune Interval	SR/configure/ip/pim> join-prune-interval <time>
Configure MRT Period	SR/configure/ip/pim> mrt-period <time>
Configure MRT Stale Multiplier	SR/configure/ip/pim> mrt-stale-mult <number>
Configure MRT SPT Multiplier	SR/configure/ip/pim> mrt-spt-multiplier <number>
Configure Probe Period	SR/configure/ip/pim> probe-period <time>
Configure Registration suppression timeout	SR/configure/ip/pim> register-suppress-timeout <time>
Configure DR to switch immediate	SR/configure/ip/pim> dr-switch-immediate
Configure RP to switch immediate	SR/configure/ip/pim> rp-switch-immediate
Configure Threshold for DR	SR/configure/ip/pim> threshold-dr <bps>
Configure Threshold for RP	SR/configure/ip/pim> threshold-rp <bps>
Configure to calculate whole packet checksum (for cisco interop)	SR/configure/ip/pim> whole-packet-checksum
Bootstrap Router related Commands	
Configure as candidate BSR	SR/configure/ip/pim/cbsr> address <address>
Configure CBSR period	SR/configure/ip/pim/cbsr> period <time>
Configure CBSR holdtime	SR/configure/ip/pim/cbsr> holdtime <time>
Configure CBSR interface	SR/configure/ip/pim/cbsr> interface <interface name>
Configure CBSR priority	SR/configure/ip/pim/cbsr> priority <value>
RP commands	
Configure as candidate RP	SR/configure/ip/pim> crp
Configure as candidate RP address	SR/configure/ip/pim/crp> address <ipaddress>
Configure candidate RP group for advertisement	SR/configure/ip/pim/crp> group-add <address> [mask] [priority]
Configure as candidate RP holdtime	SR/configure/ip/pim/crp> holdtime <time>
Configure as candidate RP interface	SR/configure/ip/pim/crp> interface <interface name>

Configure as candidate RP period	SR/configure/ip/pim/crp> period <time>
Configure as candidate RP priority	SR/configure/ip/pim/crp> priority <value>
Configure a static RP address	SR/configure/ip/pim/> rp <address> <gaddress> [mask]
Inteface based parameters	
Configure PIM for an interface	SR/configure/ip/pim> interface <interface_name>[:dlci_no]
Configure PIM mode for an interface	SR/configure/ip/pim/interface wan1> mode [sparse | dense | ssm | sparse-ssm]
Configure PIM interface hello holdtime	SR/configure/ip/pim/interface wan1> hello-holdtime <time>
Configure PIM interface hello interval	SR/configure/ip/pim/interface wan1> hello-interval <time>
Configure PIM interface hello priority	SR/configure/ip/pim/interface wan1> hello-priority <time>
Configure PIM interface Join/ Prune Interval	SR/configure/ip/pim/interface wan1> join-prune-interval <time>
Configure PIM interface Join/ Prune holdtime	SR/configure/ip/pim/interface wan1> join-prune-holdtime <time>
Configure PIM interface as border of PIM domain	SR/configure/ip/pim/interface wan1> boundary
SSM range	
Configure the SSM range	SR/configure/ip/pim> ssm-range <group-address> <group-mask

The show and debug PIM commands are:

Display PIM global configuration	SR> show ip pim global
Display PIMC timers	SR> show ip pim timers
Display PIM interfaces	SR> show ip pim interface /all
Display PIM neighbors	SR> show ip pim neighbors
Display PIM Bootstrap info	SR> show ip pim bsr-info
Display PIM Candidate RP info	SR> show ip pim crp-info
Display PIM statistics	SR> show ip pim statistics

Display PIM RP set	SR> show ip pim rp-set
Display PIM Static RP	SR> show ip pim rp
Trace PIM alert	SR> debug ip pim alert
Trace PIM detail	SR> debug ip pim detail
Trace PIM packets	SR> debug ip pim packet <pkt_type> direction [level] <detail> [dlci]
Trace PIM state changes	SR> debug ip pim state
Trace PIM routes	SR> debug ip pim route

Protocol Independent Multicast - Source Specific Multicast (PIM-SSM)

By running PIM-SSM and IGMPv3, you can implement a Source Specific Multicast (SSM) service model in your network. PIM-SSM functionality is the subset of PIM-SM functionality dealing only with source-specific distribution trees. IGMPv3 provides a way to detect channel subscriptions; for example, host-initiated (S,G) joins where G falls within the defined range of SSM multicast group addresses.

PIM-SSM can be run in the absence of IGMPv3. Even when they are both running, they do not need to run on the same interfaces. For example, you might not want to run PIM-SSM (or PIM-SM, for that matter) over an interface that leads to a stub network. Similarly, if a network cannot contain local IGMPv3 receivers (for example, on a DMZ network), then there is no need to run IGMPv3.

PIM-SSM Command

```
ip pim ssm-range
```

Internet Group Management Protocol (IGMP)

Internet Group Management Protocol (IGMP) is enabled on hosts and routers that want to receive multicast traffic. IGMP informs locally-attached routers of their multicast group memberships. Hosts inform routers of the groups of which they are members by multicasting IGMP Group Membership Reports. When multicast routers listen for these reports, they can exchange group membership information with other multicast routers. This reporting system allows distribution trees to be formed to deliver multicast datagrams. The original version of

IGMP was defined in RFC 1112, Host Extensions for IP Multicasting. Extensions to IGMP, known as IGMP version 2.

IGMPv2 improves performance and supports the following message types:

Example

- IGMP Query: IGMP Query is sent by the router to know which groups have members on the attached network.
- IGMP Reports: IGMP reports are sent as a response to the query by hosts to announce their group membership. Reports can be sent "unsolicited" when the hosts come up.
- IGMP Leaves: IGMP Leaves are sent by the host when it relinquishes membership of a group.

The latest extension to the IGMP standard is Version 3, which includes interoperability with version 2 and version 1 hosts, also provides support for source filtering. Source filtering enables a multicast receiver host to signal to a router which groups it wants to receive multicast traffic from, and from which source(s) this traffic is expected. This membership information enables the router to forward traffic only from those sources from which receivers requested the traffic.

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast host group in the following two modes:

Example

- INCLUDE mode: In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.
- EXCLUDE mode: In this mode, the receiver announces membership to a host group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. This indicates that the host wants to receive traffic only from other sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

IGMPv3 is used by the hosts to express their desire to be a part of the source-specific multicast (SSM) which is an emerging standard used by routers to direct multicast traffic to the host only if its is from a specific source.

IGMP Commands

The IGMP commands supported are:

Enabling igmp	SR/configure > ip igmp
Disabling igmp	SR/configure > no ip igmp

Enabling igmp	SR/configure/ip/igmp> interface <interface-name>[: dlci-no]
Disabling igmp	SR/configure/ip/igmp> no interface <interface-name>
Configuring version	SR/configure/ip/igmp/interface ethernet0> version <version>
Configuring Query Interval	SR/configure/ip/igmp/interface ethernet0> query-interval <interval>
Configuring Maximum Response Time	SR/configure/ip/igmp/interface ethernet0> query-response-interval <interval>
Configuring Last Member Query Interval	SR/configure/ip/igmp/interface ethernet0> last-member-query-interval <interval>
Configuring Last Member Query Count	SR/configure/ip/igmp/interface ethernet0> last-member-query-count <value>
Configuring Startup Query Interval	SR/configure/ip/igmp/interface ethernet0> startup-query-interval <interval>
Configuring Startup Query Count	SR/configure/ip/igmp/interface ethernet0> startup-query-count <count>
Configuring Robustness	SR/configure/ip/igmp/interface ethernet0> robustness <value>
Configuring Ignore-v1-message	SR/configure/ip/igmp/interface ethernet0> [no] ignore-v1-messages
Configuring Ignore-v2-message	SR/configure/ip/igmp/interface ethernet0> [no] ignore-v2-messages
Configuring Send Router Alerts	SR/configure/ip/igmp/interface ethernet0> [no] send-router-alert
Configuring Require Router Alerts	SR/configure/ip/igmp/interface ethernet0> [no] require-router-alert
Debug Command	
Debug state related events	SR/debug> [no] ip igmp state
Debug normal events	SR/debug> [no] ip igmp normal
Debug all packets	SR/debug> [no] ip igmp packet all [inbound | outbound]
Debug leave packets	SR/debug> [no] ip igmp packet leave [inbound | outbound]
Debug query packets	SR/debug> [no] ip igmp packet query [inbound | outbound]
Debug report packets	SR/debug> [no] ip igmp packet report [inbound | outbound]
Show Commands	
Displaying IGMP group membership information	SR/ show ip igmp groups {all | <interface-name>} [detail]
Displaying IGMP interface configuration	SR/show ip igmp interface {all | <interface-name>}

Clear Command	
Clearing IGMP group membership information	SR/clear ip igmp groups [interface <name>] [group-addr <addr>] [source-addr <source-addr>]

Traceroute Facility for IP Multicast

With multicast distribution trees, tracing from a source to a multicast destination is difficult, since the branch of the multicast tree on which the destination lies is unknown. The technique used by the traceroute tool to trace unicast network paths will not work for IP multicast because traceroute (ICMP) responses are specifically forbidden for multicast traffic. Thus, you have to flood the whole tree to find the path from one source to one destination. However, walking up the tree from destination to source is easy, as most existing multicast routing protocols know the previous hop for each source. Tracing from destination to source involves only routers on the direct path.

To request a traceroute (which does not have to be the source or the destination), send a traceroute query packet to the last-hop multicast router for the given destination. The last-hop router turns the query into a request packet by adding a response data block containing its interface addresses and packet statistics, and then forwards the request packet using unicast to the router that it believes is the proper previous hop for the given source and group. Each hop adds its response data to the end of the request packet, then unicast forwards it to the previous hop. The first hop router (the router that believes that packets from the source originate on one of its directly connected networks) changes the packet type to indicate a response packet and sends the completed response to the response destination address. The response may be returned before reaching the first hop router if a fatal error condition such as "no route" is encountered along the path.

Multicast traceroute uses any information available to it in the router to try to determine a previous hop to forward the trace towards. Multicast routing protocols vary in the type and amount of state they keep; multicast traceroute tries to work with all of them by using whatever is available. For example, if a DVMRP router has no active state for a particular source but does have a DVMRP route, it chooses the parent of the DVMRP route as the previous hop. If a PIM-SM router is on the (*,G) tree, it chooses the parent towards the RP as the previous hop. In these cases, no source/group-specific state is available; the path may still be traced.

This release supports the following PIM related feature—a "traceroute" facility for IP multicast, as defined in draft-ietf-idmr-traceroute-ipm-05.

Multicast Multipath

The multicast multipath feature allows load balancing on multicast traffic across equal cost paths. Equal cost multipath routing is useful when multiple equal cost routes to the same destination exist. These routes can be discovered and be used to provide load balancing among redundant paths. Commonly used methods for multipath forwarding are Round-Robin and Random. While these methods do provide a form of load balancing, but variable path MTUs, variable latencies, and debugging can limit the effectiveness of these methods.

The following methods have been developed to deal with the load balancing limitations of the Round-Robin and Random methods:

- Modulo-N Hash —To select a next-hop from the list of N next-hops, the router performs a modulo-N hash over the packet header fields that identify a flow."
- Hash-Threshold —The router first selects a key by performing a hash over the packet header fields that identify the flow. The N next-hops have been assigned unique regions in the hash functions output space. By comparing the hash value against region boundaries the router can determine which region the hash value belongs to and thus which next-hop to use.
- Highest Random Weight (HRW) —The router computes a key for each next-hop by performing a hash over the packet header fields that identify the flow, as well as over the address of the next-hop. The router then chooses the next-hop with the highest resulting key value.

The Round-Robin and Random methods are disruptive by design (that is, if there is no change to the set of next-hops, the path a flow takes changes every time). Modulo-N, Hash Threshold, and HRW are not disruptive.

RFC 2991 recommends to use HRW method to select the next-hop for multicast packet forwarding. or this reason, Avaya-only scenarios apply the HRW method as the default. This is similar to the Cisco Systems IPv6 multicast multipath implementation.

Multipath Commands

The following table lists the multipath commands:

Enabling HRW method	SR/configure/ip/multicast> multipath
Enabling Cisco method	SR/configure/ip/multicast> multipath cisco
Disabling Multipath	SR/configure/ip/multicast> no multipath SR/configure/ ip/multicast> no multipath cisco
Display RPF selection	SR> show ip rpf <addr> <addr> - source or RP address

When multipath is disabled, the Secure Router selects the nexthop address with lowest ip address. For equal cost routes the nexthops are stored in the increasing (ascending) order of IP address. show ip rpf command displays the selected path, based on the configured multipath method and the nexthops of the best route to the IP address passed.

Generic Routing Encapsulation (GRE)

Avaya's implementation of the Generic Routing Encapsulation (GRE) tunneling protocol is based on standards RFC1701 and RFC2784.

GRE can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link between routers at remote points over an IP network. An IP tunnel is a logical interface that provides a way to encapsulate passenger packets inside a transport protocol. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling using GRE allows network expansion across a single-protocol backbone environment.

For more information on GRE, refer to the *SR1000 Configuration Guide* .

Chapter 15: Secure Router Configuration for Dynamic Route Exchange over IPSec Tunnel interoperability with VPN Router

Both Avaya Secure Router 1000 series and VPN router currently support dynamic routing over IPSec. Secure router 1000 series configuration for dynamic route exchange over IPSec Tunnel allows interoperability by using IP-on-IP over a transport mode IPSec connection.

Capabilities

Secure router configuration for dynamic route exchange over IPSec Tunnel, has the following capabilities:

- IPSec transport mode is used, not tunnel mode
- The Secure Router default IPIP tunnel MTU needs to be set to 1500 for OSPF, to match the VPN Router tunnel MTU..
- If both “ip mtu” and “tunnel path-mtu-discovery” are configured/ enabled on Secure Router the mtu value set by “ip mtu” configuration will be in effect.

Secure router configuration for BGP

Configure secure routing for BGP as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
ip address 100.1.1.1 24
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
tunnel mode ipip
tunnel protection toCes Avaya
crypto untrusted
interface loopback ll1
ip address 50.1.1.1 24
exit

router ospf
redistribute connected
exit
router bgp 100
neighbor 50.1.1.10 100
update_source 50.1.1.1
exit
exit

interface loopback ll1
ip address 50.1.1.1 24
exit
Ip route 50.1.1.10 32 toCes
```

Secure router configuration for OSPF

Configure secure routing for OSPF as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
Ip unnumbered ethernet1
ip mtu 1500
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
tunnel mode ipip
tunnel protection toCes Avaya
crypto untrusted
exit

router ospf
interface toCes area 0
exit
```

Secure router configuration for RIPv2

Configure secure routing for RIPv2 as follows:

```
interface ethernet 0
ip address 10.10.10.1 24
```

```
crypto trusted
exit

interface ethernet 1
ip address 192.168.26.100 24
crypto untrusted
exit

interface tunnel toCes
ip address 100.1.1.1 24
tunnel source 192.168.26.100
tunnel destination 192.168.27.100
tunnel mode ipip
tunnel protection toCes Avaya
crypto untrusted
exit

ip route 192.168.27.0 24 192.168.26.101
router routerid 192.168.26.100
router rip
interface toCes
mode 2
exit

firewall corp
policy 101 in
exit
exit

firewall internet
policy 100 in self
exit
exit
```