Foundry AR-Series Router User Guide

For AR1202, AR1204, AR1208, AR1216, AR3201-CH/CL, and AR3202-CH/CL Routers

2100 Gold Street
P.O. Box 649100
San Jose, CA 95164-9100
Tel 408.586.1700
Fax 408.586.1900

June 2004

Contents

CHAPTER 7
BGP4 CONFIGURE COMMANDS 7-1

CHAPTER 8
BGP4 SHOW COMMANDS 8-1

Chapter 1
Getting Started

Introduction

This guide describes how to configure the AccessIron routers in typical scenarios using information presented in the configurations and user guides.

Audience

This manual is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Foundry Layer 3 Switch, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, BGP4, PIM, and VRRP.

Nomenclature

This guide uses the following typographical conventions to show information:

Italic	highlights the title of another publication and occasionally emphasizes a word or phrase.
Bold	highlights a CLI command.
Bold Italic	highlights a term that is being defined.
Underline	highlights a link on the Web management interface.
Capitals	highlights field names and buttons that appear in the Web management interface.

NOTE: A note emphasizes an important fact or calls your attention to a dependency.

WARNING: A warning calls your attention to a possible hazard that can cause injury or death.

CAUTION: A caution calls your attention to a possible hazard that can damage equipment.

Related Publications

The following Foundry Networks documents supplement the information in this guide.

- Release Notes

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

- *Foundry AR-Series AR1202 and AR1204 Installation Guide*

  This guide is designed to assist users with the initial installation and deployment of the Foundry AR1202 two-port and AR1204 four-port router. The guide provides a brief overview of the installation and initial configuration processes.

- *Foundry AR-Series AR1202 and AR1204 Quick Installation Guide*

  This detailed guide provides an abbreviated install guide for those experienced with installing Foundry AccessIron routers.

- *Foundry AR-Series Rack-Mounted Router Installation Guide*

  This guide is designed to assist users with the initial installation and deployment of Foundry rack-mounted routers. The guide provides a brief overview of the installation and initial configuration processes.

- *Foundry AR-Series Rack-Mounted Router Quick Installation Guide*

  This detailed guide provides an abbreviated install guide for those experienced with installing Foundry AccessIron rack-mounted routers.

- *Foundry AR-Series Router Configurations Guide*

  This guide provides examples of AccessIron configurations.

- *Foundry AR-Series Router Command Reference Guide*

  This guide explains the syntax and application of AccessIron router CLI commands.

To order additional copies of these manuals, do one of the following:

- Call 1.877.TURBOCALL (887.2622) in the United States or 1.408.586.1881 outside the United States.
- Send email to info@foundrynet.com.

List of Features

Table 1.1 shows the features supported on AccessIron devices.

Table 1.1: Feature Supported in AccessIron Devices

Category	Feature	AR1202 AR1204 AR1208 AR1216	AR3201-T-CL AR3202-T-CL	AR3201-T-CH AR3202-T-CH
Interfaces				
WAN/LAN	10/100 Fast Ethernet	2	2	2
	T1/E1	Yes	-	-
	Channelized T3	-	-	Yes
	Clear Channel T3	-	Yes	-
WAN Protocols				

Table 1.1: Feature Supported in AccessIron Devices (Continued)

Category	Feature	AR1202 AR1204 AR1208 AR1216	AR3201-T-CL AR3202-T-CL	AR3201-T-CH AR3202-T-CH
PPP, PAP, Multilink PPP, Frame Relay, Multilink Frame Relay, (FRF.15, FRF.16.1) BCP, HDLC				
Layer 2 Features				
	802.1Q VLAN tagging and forwarding over WLAN			
	Virtual LAN Domain (VLD) VLAN Double Tagging			
	Transparent Bridging			
	Jumbo Frames (4072 bytes)			
	IP Multiplexing			
	NAT mode			
	Transparent Layer 3 packet forwarding			
Layer 3 Features				
Routing	RIPv1/v2			
	OSPF			
	BGP4			
	Static Routing			
	ECMP (IP load balancing)			
	Multicast (PIM-SM, PIM-SSM, IGMP v2/v3)			
High Availability	VRRP			
	BGP4 Multi-homing			
	Bundle Tracking			
	MLPPP Bundle Thresholding			
	LAN Interface Load Sharing with Failover			
Security/ Management	Stateful Packet Inspection Firewall with: Layer-3 mode (router and NAT) Policy-based NAT/PAT Policy-based filters URL and application content filtering Time and rate limiting Denial of Service protection Network attack detection Application Level Gateway support Packet-level logging and syslog support			

Table 1.1: Feature Supported in AccessIron Devices (Continued)

Category	Feature	AR1202 AR1204 AR1208 AR1216	AR3201-T-CL AR3202-T-CL	AR3201-T-CH AR3202-T-CH
	ACLs			
	DHCP			
	TFTP			
	PAP			
	RADIUS			
	TACACS+			
	SSH v2			
	GRE Tunneling			
	IPSec VPN with integrated IKE Site-to-site VPN Site-to-remote VPN MD5 & SHA-1 authentication Hardware accelerated encryption 3DES (168 bit), DES (56 bit), AES (256 bit) encryption	VPN optional on the AR1202 and AR1204	-	-
QoS/Traffic Management	RED			
	DiffServ			
	Class-based Queuing per: IP address Flow VLAN tag Application port			
	Frame Relay traffic shaping and policing			
	VLAN-802.1P 8 queue prioritization of VLAN frames			
Service Provisioning	Management (in-band, serial, Telnet, or modem) by: CLI SNMP			
	Monitoring syslog Statistics Alarms			
	Diagnostics BERT Loopback testing Traceroute Reverse Telnet			
Specialized Features	Hospitality Web Redirection			

Table 1.1: Feature Supported in AccessIron Devices (Continued)

Category	Feature	AR1202 AR1204 AR1208 AR1216	AR3201-T-CL AR3202-T-CL	AR3201-T-CH AR3202-T-CH
	Timed Access List			

How to Get Help

Foundry Networks technical support will ensure that the fast and easy access that you have come to expect from your Foundry Networks products will be maintained.

Web Access

- http://www.foundrynetworks.com

Email Access

Technical requests can also be sent to the following email address:

- support@foundrynet.com

Telephone Access

- 1.877.TURBOCALL (887.2622) United States
- 1.408.586.1881 Outside the United States

Warranty Coverage

Contact Foundry Networks using any of the methods listed above for information about the standard and extended warranties.

Chapter 2
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 system 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 systems 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 system. 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 systems 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 2.1: shows a context-sensitive command string for a AR1208 system. In this example, T1 link 1 remains selected for configuration until you exit from the Foundry-AR1208/configure/module/t1# prompt.

Table 2.1: Context-Sensitive Command Sequence

Context-Sensitive Command String	Example
1 Go into the **configuration** mode.	Foundry-AR1208#**configure terminal**
1 Specify the type of interface (T1).	Foundry-AR1208/configure#**module t1**
1 Choose the specific interface (T1 link 1).	Foundry-AR1208/configure#**module t1 1**
1 From now on, all configuration commands are for T1 link 1 until you exit from module configuration or choose another T1 link.	Foundry-AR1208/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 2.2: 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: connections connections < n > Command execution: connections 4 In other words, the first occurence of “connections” must be entered because it is not followed by a setting enclosed in angled brackets. The second occurence 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.

Table 2.2: Syntax Conventions (Continued)

[a | b | c]	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: timeout protocol_type < tcp | udp > [seconds < n >] Command execution: timeout udp 3600
< #	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.
[ ]	Optional parameter settings in each syntax are indicated by normal brackets.

Table 2.3: Example Conventions

For Examples	What it means
normal type	Prompts and commands that are part of the main prompt are shown in normal type. Examples: Foundry-AR1208# Foundry-AR1208/show#
bold type	All character strings that a user must enter to execute a command are in bold type. Example: Foundry-AR1208# **configure term**

Abbreviated Commands

You may enter commands by typing the first few characters of each word in a command string. The Foundry system 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 Foundry# prompt. You may also type **dis con run** 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 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**.

Figure 2.1 Navigation Keys

```
# 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
#
```

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 Foundry system help information. Or, type **help** followed by a command name to view information about that command. shows the help screen.

Figure 2.2 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
#
```

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 2.3 shows two command tree examples. If you type **tree** at the main (Foundry-AR1208# or equivalent) prompt, the entire list of system commands appears. If you type **tree** within a command mode, such as Foundry-AR3201-CH/clear# **tree**, the commands associated with this command mode are displayed**.**

Figure 2.3 Foundry CLI Command Tree

```
 # 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) :
```

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 system command categories appear. Shows a display of these top-level commands.

Figure 2.4 ? Help Screen

```
# ?

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
      mtrace          -- multicast trace route to source address
      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

#
```

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 2.5 on page 2-7, 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 Foundry system and other network hosts; access to the **save** commands from anywhere in the CLI ensures that your configurations may be saved periodically.

Figure 2.5 Global show Command

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

CONTENTS OF /flash1:

  size          date       time       name
--------       ------     ------    --------
 6467513    FEB-04-2004  13:51:22   AR0x_###x
 6771268    APR-01-2004  11:38:42   AR0x_###y
    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   AR0x_###z

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

NOTE: The CLI commands show and display can be used interchangeably.

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

Chapter 3
Policy Commands

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

configure policy

This command provides access to the next-level commands.

related 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 13-1.) Note that the regular expression must be enclosed in quotation marks. The AS number is the smallest element of a Foundry regular expression. It is an integer ranging from 0 to 65536; the Foundry 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	Description
access_list	Access list number
	Range is 1 - 199.
number	Sequence to insert or delete from an existing AS path entry.
	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 13-1 for more information about regular expressions.

syntax:

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

example:

Foundry-AR1208/configure# **policy as_path 1 120 permit “100”**

example:

Foundry-AR1208/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 contai8n 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.

related 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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/configure# **policy community_list standard_community 90 150 permit community 40 45 local_as aa_nn 655:232592 no_advertise**

example:

Foundry-AR1208/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 network. 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, maskmask) 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	Description
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:

Foundry-AR1208/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:

Foundry-AR1208/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]

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	Description
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:

Foundry-AR1208/configure# **policy route_map Block100 1 permit**

related 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.

related 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	Description
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:

Foundry-AR1208/configure#/policy/route_map Block100 1# **match as_path 1**

related 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:

Foundry-AR1208/configure/policy/route_map Block100 1# **match community**

related 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	Description
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:

Foundry-AR1208/configure/policy/route_map Block100 1# **match ip ip_address 20**

applicable systems:

All models.

configure policy route_map set

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

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

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	Description
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:

Foundry-AR1208/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
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	Descriptiongenerate_
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.
gemerate_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:

Foundry-AR1208/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	Description
distance	Default preference value The range is 0 - 255.

syntax:

[no] set distance distance < n >

example:

Foundry-AR1208/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

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	Description
local_preference	Preference value The range is 1 - 4292967294.

syntax:

[no] set local_preference local_preference < n >

example:

Foundry-1450configure/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	Description
metric	Metric value The range is 1 - 4294967294.

syntax:

[no] set metric metric < n >

example:

Foundry-AR1208/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	Description
type	Internal
internal	Use the IGP metric as the MED for BGP.

syntax:

[no] set metric_type type < internal >

example:

Foundry-AR1208/configure/policy/route_map Block100 1# **set metric_type internal**

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

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	Description
origin	
egp	EGP protocol
igp	IGP protocol
incomplete	Unknown protocol type

syntax:

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

example:

Foundry-AR1208/configure/policy/route_map Block100 1# **set origin igp**

applicable systems:

All models.

related commands:

configure policy route_map set origin egp

configure policy route_map set origin igp

configure policy route_map set origin incomplete

Chapter 4
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 Foundry Network's BGP RFC compliance information.

Table 4.1: 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 Foundry' 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 Foundry Network's OSPF RFC compliance information.

Table 4.2: 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 required to determine a next hop and bandwidth could 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 Foundry' 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
- Inbound and outbound filtering policies

RFC Compliance

The following table provides Foundry Network's RIP RFC compliance information.

Table 4.3: RIP RFC Compliance

RFC	Description
1058	Routing Information Protocol
2453/ STD0056	RIP Version 2
1724	RIP Version 2 MIB extension

Table 4.3: RIP RFC Compliance

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). Foundry 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

This feature allows AR-series router 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 5
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

related commands:

clear ip bgp all

clear ip bgp group

clear ip bgp neighbor

example:

Foundry-AR1208# **clear ip bgp**

applicable systems:

All models.

clear ip bgp all

This command removes all BGP neighbor connections.

syntax:

clear ip bgp all

example:

Foundry-AR1208# **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	Description
group_name	Name of the group

syntax:

clear ip bgp group group_name < name >

example:

Foundry-AR1208# **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	Description
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:

Foundry-AR1208# **clear ip bgp neighbor 10.1.1.1 200**

related commands:

clear ip bgp all

clear ip bgp group

applicable systems:

All models.

Chapter 6
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 Foundry.

configure router

This command provides access to next-level commands.

related commands:

configure router routerid

configure router routerid

This command configures a router for routing operation.

syntax:

[no] router routerid < *IP address#*

example:

Foundry-AR1208/configure# **router routerid 10.10.10.10**

applicable systems:

All models.

show ip routes

This command displays IP routing information for Ethernet ports.

Parameter	Description
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	
rib	RIB routes
fib	FIB routes

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 6.1: 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:

Foundry-AR1208/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 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	Description
as_number	The number of an autonomous system. The range is 1 - 65535.

syntax:

[no] router bgp as_number < n >

example:

Foundry-AR1208/configure# **router bgp 10**

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

applicable systems:

All models.

configure router bgp aggregate_address

This command is used to aggregate routes.

Parameter	Description
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 network < *IP address* > mask < *subnet mask* > [generate_as_set < as_set >] [generate_summary_only < summary_only >] [suppress_map < name >] [advertise_map < name >] [attribute_map < name >]

example:

Foundry-AR1208/configure/router/bgp 10# **aggregate_address 100.3.0.0 255.255.0.0**

related commands:

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

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:

Foundry-AR1208/configure/router/bgp 10# **always_compare_med**

related commands:

configure router bgp aggregate_address

configure router bgp distance

configure router bgp default_metric

configure router bgp group

configure router bgp neighbor

configure router bgp 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	Description
default_metric	The default metric value. The range is 1 - 4294967294.

syntax:

[no] default_metric default_metric < n >

example:

Foundry-AR1208/configure/router/bgp 10# **default_metric 2000**

related commands:

configure router bgp aggregate_address

configure router bgp always_compare_med

configure router bgp distance

configure router bgp group

configure router bgp neighbor

configure router bgp redistribute

applicable systems:

All models.

configure router bgp distance

This command changes the default distance value on a router.

Higher values are preferred.

Parameter	Description
distance	Default preference value The range is 0-255; the default is 170.

syntax:

[no] distance distance < n >

example:

Foundry-AR1208/configure/router/bgp 10# **distance 20**

Table 7.1: 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 aggregate_address

configure router bgp always_compare_med

configure router bgp default_metric

configure router bgp group

configure router bgp neighbor

configure router bgp redistribute

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	Description
name	Group name to be configured
group_type	
external	External routing group
	Default group name = FoundryBgpExternal
external_rt	External routing group
	Default group name = FoundryBgpExternalRt
internal	Internal routing group
	Default group name = FoundryBgpInternal

syntax:

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

example:

Foundry-AR1208/configure/router/bgp 10# **group toronto internal**

related 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

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	Description
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:

Foundry-AR1208/configure/router/bgp 10/group toronto internal# **distribute_list 101 out**

related commands:

configure router bgp group filter_list

configure router bgp group next_hop_self

configure router bgp group password

configure router bgp group remove_private_AS

configure router bgp group route_map

applicable systems:

All models.

configure router bgp group filter_list

This command configures BGP filters for a specified group.

Parameter	Description
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:

Foundry-AR1208/configure/router/bgp 10/group toronto internal# **filter_list 103 out**

related commands:

configure router bgp group distribute_list

configure router bgp group next_hop_self

configure router bgp group password

configure router bgp group remove_private_AS

configure router bgp 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:

Foundry-AR1208/configure/router/bgp 10/group blue external# **next_hop_self**

related commands:

configure router bgp group distribute_list

configure router bgp group filter_list

configure router bgp group password

configure router bgp group remove_private_AS

configure router bgp 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	Description
md5_password	TCP MD5 password (string) for the group Enter a word.

syntax:

[no] password md5_password < string >

example:

Foundry-AR1208/configure/router/bgp 10/group toronto internal# **password rt56htd**

related commands:

configure router bgp group distribute_list

configure router bgp group filter_list

configure router bgp group next_hop_self

configure router bgp group remove_private_AS

configure router bgp 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:

Foundry-AR1208/configure/router/bgp 10/group toronto internal# **remove_private_AS**

related 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

configure router bgp 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	Description
route_map	Route map name
route_map_options	
out	Outbound direction

syntax:

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

example:

Foundry-AR1208/configure/router bgp 10/group toronto internal# **route_map foo out**

related 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

configure router bgp group remove_private_AS

applicable systems:

All models.

configure router bgp neighbor

This command configures a BGP neighbor.

Parameter	Description
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:

Foundry-AR1208/configure/router/bgp 10# **neighbor 101.101.1.2 4**

related commands:

configure router bgp neighbor advertisement_interval
configure router bgp neighbor allowbadid
configure router bgp neighbor default_originate
configure router bgp neighbor description
configure router bgp neighbor distribute_list
configure router bgp neighbor ebgp_multihop
configure router bgp neighbor filter_list
configure router bgp neighbor keep
configure router bgp neighbor logupdown
configure router bgp neighbor maximum_prefix
configure router bgp neighbor neighbor_group
configure router bgp neighbor next_hop_self
configure router bgp neighbor password
configure router bgp neighbor route_map
configure router bgp neighbor timers
configure router bgp neighbor update_source

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 redistribute

applicable systems:

All models.

configure router bgp neighbor advertisement_interval

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

Parameter	Description
advertisement_interval	Time, in seconds The range is 1 - 600 seconds.

syntax:

[no] advertisement_interval advertisement_interval < n >

example:

Foundry-AR1208/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:

Foundry-AR1208/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	Description
route_map	The name of the route map

syntax:

[no] default_originate [route_map < name >]

example:

Foundry-AR1208/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	Description
neighbor_description	Text string in quotes describing neighbor

syntax:

[no] description neighbor_description < “*string*” >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/configure/router/bgp 10/neighbor 101.101.1.2 4# **filter_list 103 in**

applicable systems:

All models.

configure router bgp neighbor keep

This command configures neighbor route storage options.

Parameter	Description
keep_option	
all	Keep all non-active routes
none	Don't store non-active routes

syntax:

keep keep_option < all | none >

example:

Foundry-AR1208/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:

Foundry-AR1208/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	Description
prefix_number	Maximum prefix limit The range is 1 - 1000000.

syntax:

maximum_prefix prefix_number < n >

example:

Foundry-AR1208/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	Description
neighbor_group	The name of a neighbor group.

syntax:

[no] neighbor_group neighbor_group < name >

example:

Foundry-AR1208/configure/router/bgp 10/neighbor 101.101.1.2 4# **neighbor_group internal-group**

applicable systems:

All models.

configure router bgp neighbor next_hop_self

This command disables the next hop calculation for this neighbor.

syntax:

next_hop_self

example:

Foundry-AR1208/configure/router/bgp 10/neighbor 10.10.20.1 2# **next_hop_self**

applicable systems:

All models.

configure router bgp neighbor password

This command configures a password for md5 authentication.

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

syntax:

md5_password < string >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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	Description
keepalive	The keepalive interval The range is 2 - 21845; the default is 60.

syntax:

[no] timers keepalive < n >

example:

Foundry-AR1208/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:

Foundry-AR1208/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.

related 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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/configure/router/bgp 10# **redistribute ospf metric AR1208**

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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/configure/router/bgp 10# **redistribute static metric 25**

related commands:

configure router bgp redistribute connected

configure router bgp redistribute ospf

configure router bgp redistribute rip

applicable systems:

All models.

Chapter 8
BGP4 show Commands

Use BGP show commands to display all configured BGP information.

NOTE: The CLI commands “show” and “display” can be used interchangeably.

show ip bgp

This command accesses the following next-level display (show) commands.

related 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	Description
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:

Foundry-AR1208# **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	Description
number	Community number (enter a list of unsigned numbers)
	The maximum list size is 10.
	The range is 1 - 4294967294
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:

Foundry-AR1208# **show ip bgp community aa:nn 0:999**

Table 8.1: 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:

Foundry-AR1208# **show ip bgp groups north**

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

syntax:

show ip bgp neighbors [group < name >] [address < *IP address#*]
[routes < advertised_routes | received_routes >]

example:

Foundry-AR1208# **show ip bgp neighbors**

Table 8.2: 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 8.3: 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 remote 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 notifications

Table 8.3: Other BGP show Descriptions (Continued)

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:

Foundry-AR1208# **show ip bgp paths**

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

Table 2 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 3 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	Description
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:

Foundry-AR1208# **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:

Foundry-AR1208# **show ip bgp summary**

```
# 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 8.4: 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:

Foundry-AR1208# **show ip bgp table**

Table 8.5: 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:

related 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	Description
access_list	The access list number. The range is 1 - 199.

syntax:

show policy as_path [access_list < n >]

example:

Foundry-AR1208# **show policy as_path**

```
# 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

show policy route_map

applicable systems:

All models.

show policy community_list

This command shows configured community lists.

Parameter	Description
community	The community list number. The range is 1 - 199.

syntax:

show policy community_list [community < n >]

example:

Foundry-AR1208# **show policy community_list**

```
#show policy community_list
Community extended access list 100
   deny 0:44 ....
   permit 655: ....
```

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	Description
number	IP access list number The range is 1 - 99.

syntax:

show policy ip_access_list [number < n >]

example:

Foundry-1450/show# **policy ip_access_list**

```
# show policy ip_access_list
IP access list 1
  permit 10.0.0.0 255.255.255.255 0.255.255.255 255.255.255.255
  permit 20.0.0.0 255.255.255.255 255.255.255.255 255.255.255.255
IP access list 2
  permit 20.0.0.0 255.255.255.255 0.255.255.255 255.255.255.255
#
```

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	Description
name	The name of the route map.

syntax:

show policy route_map [< name >]

example:

Foundry-AR1208# **show policy route_map**

```
# 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

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 *Command Reference Guide: Domestic Products* 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:

  Foundry AR3201-CH/configure/interface/bundle wan1# **no ip address 2.2.2.2 24**
  Foundry AR3201-CH/configure/interface/bundle wan1# **ip address 2.2.2.2 24 type broadcast**
- Adjacencies will not form if hello_interval, dead_interval, or area_type mismatches are present.

configure router ospf

This command configures a router for OSPF routing.

syntax:

router ospf

example:

Foundry-AR1208/configure# **router ospf**

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 ref_bw
configure router ospf timers

applicable systems:

All models.

configure router ospf 1583 Compatibility

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:

Foundry-AR1208/configure/router/ospf# **1583Compatibility**

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

syntax:

area < area_id >

example:

Foundry-AR1208/configure/router/ospf# **area 0**

related 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.

related 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:

Foundry-AR1208/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:

Foundry-AR1208/configure/router/ospf/area 1# **area_type nssa**

related 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:

Foundry-AR1208/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:

Foundry-AR1208/configure/router/ospf/area 1# **area_type stub**

related 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:

Foundry-AR1208/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	Description
default_cost	Enter a number. The range is 0 - 16777215; the default is 1.

syntax:

default_cost < n >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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	Description
virtual_link	IP address for the virtual link. Enter an IP address.

syntax:

[no] virtual_link < *IP address* >

example:

Foundry-AR1208/configure/router/ospf/area 1# **virtual_link 100.10.1.5**

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 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	Description
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:

Foundry-AR1208/configure/router/ospf/area 1/virtual_link 100.10.1.5# **authentication simple Foundry**

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

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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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.

related 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.

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

syntax:

[no] distance ospf external < n >

example:

Foundry-AR1208/configure/router/ospf# **distance ospf external 25**

Table 9.1: 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	Description
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:

Foundry-AR1208/configure/router/ospf# **distance ospf non_external 25**

Table 9.2: 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	Description
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:

Foundry-AR1208/configure/router/ospf# **interface Toronto 5**

related commands:

configure router ospf 1583Compatibility
configure router ospf area
configure router ospf distance
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 redistribute
configure router ospf ref_bw
configure router ospf interface retransmit_interval
configure router ospf timers
configure router ospf interface transmit_delay

applicable systems:

All models.

configure router ospf interface authentication

This command configures the authentication type on an interface.

Parameter		Description
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:

Foundry-AR1208/configure/router/ospf/interface toBoston# **authentication md5 Foundry**

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	Description
cost	Metric cost of sending packets on a particular OSPF interface. The range is 1 - 65535; the default is computed based on the interface bandwidth.

syntax:

[no] cost < n >

example:

Foundry-AR1208/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	Description
dead_interval	Time, in seconds The range is 1- 65535; the default is 40.

syntax:

[no] dead_interval < n >

example:

Foundry-AR1208/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	Description
hello_interval	Time in seconds The default is 10; the range is 1 - 65535.

syntax:

[no] hello_interval < n >

example:

Foundry-AR1208/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
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	Description
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:

Foundry-AR1208/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
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	Description
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:

Foundry-AR1208/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	Description
poll_interval	The time, in seconds The range is 0 - 2147483647; the default is 120.

syntax:

[no] poll_interval < n >

example:

Foundry-AR1208/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

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	Description
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:

Foundry-AR1208/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
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	Description
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:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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:

Foundry-AR1208/configure/router/ospf# **redistribute**

related 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	Description
as_number	Autonomous system number The range is 1 - 65535.
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 3-8 for more information about configuring route maps.

syntax:

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

example:

Foundry-AR1208/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	Description
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 3-8 for more information about configuring route maps.

syntax:

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

example:

Foundry-AR1208/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	Description
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 3-8 for more information about configuring route maps.

syntax:

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

example:

Foundry-AR1208/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	Description
metric	OSPF default metric The range is 1 - 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 3-8 for more information about configuring route maps.

syntax:

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

example:

Foundry-AR1208/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	Description
reference_bandwidth	Reference bandwidth in Mbps The range is 1 - 4294967.

syntax:

ref_bw < n >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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.

Chapter 10
OSPF Show Commands

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

NOTE: The CLI commands “show” and “display” can be used interchangeably.

show ip ospf area

This command displays configuration information about an OSPF area.

Parameter	Description
area_id	OSPF area ID Enter either a decimal number or an IP address.

syntax:

area [area_id]

example:

Foundry-AR1208# **show ip ospf area 1**

```
# show ip ospf area_id 1


 Area 1
    Number of interfaces in this area is 0
    Area type is NORM
```

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:

Foundry-AR1208# **show ip ospf database**

related 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	Description
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:

Foundry-AR1208# **show ip ospf database all**

```
# show ip ospf database all


              Router LSAs for Area 0
```

related 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	Description
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:

Foundry-AR1208# **show ip ospf database asbr_summary**

related 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:

Foundry-AR1208# **show ip ospf database database_summary**

```
# show ip ospf database database_summary

Area ID        Router  Network Sum-Net Sum-ASBR NSSA   Subtotal
Deleted Maxaged

-------        ------  ------- ------- -------- ----   -------- -
------ -------
```

related 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	Description
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:

Foundry-AR1208# **show ip ospf database external**

related 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	Description
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:

Foundry-AR1208# **show ip ospf database network**

related 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 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	Description
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:

Foundry-AR1208# **show ip ospf database nssa_external**

related 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 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	Description
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:

Foundry-AR1208# **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
```

related 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 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	Description
area_id	OSPF area ID Enter either a decimal number or an IP address.

syntax:

database self_originate [area_id < *n* >]

example:

Foundry-AR1208# **show ip ospf database self_originate**

```
  # show ip ospf database self_originate


              Router LSAs for Area 0
```

related 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 summary

applicable systems:

All models.

show ip ospf database summary

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

Parameter	Description
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:

Foundry-AR1208# **show ip ospf database summary**

related 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:

Foundry-AR1208# **show ip ospf global**

```
# show ip ospf global


Routing Process 'ospf 30583' with ID 10.1.1.1
It is rfc1583 incompatible
Summary Link update interval is 1800
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Reference bandwidth 100 Megabits per second
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Area 0
    Number of interfaces in this area is 0
```

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:

Foundry-AR1208# **show ip ospf interface**

related 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:

Foundry-AR1208# **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:

Foundry-AR1208# **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:

Foundry-AR1208# **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:

Foundry-AR1208# **show ip ospf neighbor**

related 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:

Foundry-AR1208# **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:

Foundry-AR1208# **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:

Foundry-AR1208# **show ip ospf neighbor interface ethernet 1**

related 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:

Foundry-AR1208# **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:

Foundry-AR1208# **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:

Foundry-AR1208# **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:

Foundry-AR1208# **show ip ospf request_list 10.10.10.1**

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 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:

Foundry-AR1208# **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:

Foundry-AR1208# **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 11
RIP Configure Commands

Use RIP configure commands to configure all RIP parameters.

NOTE: See the command **configure interface loopback** in the *Command Reference Guide: Domestic Products* for important information about loopback interfaces.

configure router rip

This command enables the Routing Information Protocol (RIP).

syntax:

[no] router rip

example:

Foundry-AR1208/configure# **router rip**

related 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	Description
metric	Default metric The range is 1 - 4294967294; the default is 1.

syntax:

[no] default_metric < n >

example:

Foundry-AR1208/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	Description
distance	Distance value (enter a number) The range is 1 - 255; the default is 100.

syntax:

distance < n >

example:

Foundry-AR1208/configure/router/rip# **distance 25**

Table 11.1: 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	Description
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:

Foundry-AR1208/configure/router/rip# **interface ethernet0**

This example configures the Ethernet 0 interface for RIP.

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

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	Description
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:

Foundry-AR1208/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	Description
access_list	Access list number
	Enter a number.
direction	Traffic flow direction
in	Inbound
out	Outbound

syntax:

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

example:

Foundry-AR1208/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	Description
metric	Default metric The range is 1 - 4294967294; the default is 1.

syntax:

[no] metric < n >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/configure/router/rip/interface ethernet0# **mode 1**

This example configures interface Ethernet 0 for RIP version 1.

related commands:

configure router rip interface authentication

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	Description
ip_address	Neighbor IP address

syntax:

[no] neighbor < ip_address >

example:

Foundry-AR1208/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:

Foundry-AR1208/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	Description
splitval	Split horizon algorithm
none	Disables split horizon.
simple	Enables split horizon.
poison	Enables poison reverse (default)

syntax:

[no] split_horizon < none | simple | poison >

example:

Foundry-AR1208/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	Description
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:

Foundry-AR1208/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:

Foundry-AR1208/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:

Foundry-AR1208/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.

related commands:

configure router rip redistribute bgp

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	Description
as_number	Autonomous system number The range is 1 - 65535.
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

redistribute bgp as_number [metric < n >]

example:

Foundry-AR1208/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	Description
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

[no] redistribute connected [metric < n >]

example:

Foundry-AR1208/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	Description
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

[no] redistribute ospf [metric < n >]

example:

Foundry-AR1208/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	Description
metric	Default metric The range is 1 - 16; the default is 1.

syntax:

[no] redistribute static [metric < n >]

example:

Foundry-AR1208/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.

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

syntax:

[no] flush time < n >

example:

Foundry-AR1208/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	Description
time	Holddown timer value in seconds The range is 1- 65535; the default is 180.

syntax:

[no] holddown time < n >

example:

Foundry-140/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	Description
time	Update timer in seconds The range is 1 - 65536; the default is 120.

syntax:

[no] update time < n >

example:

Foundry-AR1208/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.

Chapter 12
RIP show Commands

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

NOTE: The CLI commands “show” and “display” can be used interchangeably.

show ip rip

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

related 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:

Foundry-AR1208# **show ip rip global**

```
# show ip rip global
Router RIP is enabled
        Mode: RIP 2
        Distance: 100
        Default Metric: 1
        Timers:
                Update: 30 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.

related 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:

Foundry-AR1208# **show ip rip interface all**

```
# show ip rip interface all
RIP is configured for interface <ethernet0#
        Mode: RIP 2
        Metric: 5
        Authentication: None
        Split Horizon: Poison
        Routers : None
```

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	Description
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:

Foundry-AR1208# **show ip rip interface bundle Dallas**

related commands:

show ip rip interface all

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:

Foundry-AR1208# **show ip rip interface ethernet0**

```
# show ip rip interface ethernet 0
RIP is configured for interface <ethernet0#
        Mode: RIP 2
        Metric: 5
        Authentication: None
        Split Horizon: Poison
        Routers : None
```

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:

Foundry-AR1208# **show ip rip interface statistics**

```
  # show ip rip interface statistics


          RIP Interface Statistics:

          =========================


          Interface: <ethernet0#

                  Number of bad packets received   : <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:

Foundry-AR1208# **show ip rip statistics**

```
 Show ip rip statistics


         RIP Global Statistics:
         ==============================


         Number of Global Route Changes : <0#
```

related commands:

show ip rip global

show ip rip interface

show ip rip routes

applicable systems:

All models.

Chapter 13
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 that spaces are ignored in general, but should be used between two concatenated ASs to distinguish 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 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). Foundry 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:

TABLE 4 PIM COMMANDS

Global parameters	
Enable PIM	Foundry/configure/ip# pim
Configure PIM mode	Foundry/configure/ip/pim# mode [sparse | dense]
Configure Assert Holdtime	Foundry/configure/ip/pim#assert-holdtime <time#
Configure Hello Interval	Foundry/configure/ip/pim#hello-interval <time#
Configure Hello Holdtime	Foundry/configure/ip/pim#hello-holdtime <time#
Configure Hello priority	Foundry/configure/ip/pim#hello-priority <value#

TABLE 4 PIM COMMANDS (CONTINUED)

Configure Join/Prune Holdtime	Foundry/configure/ip/pim#join-prune-holdtime <time#
Configure Join /Prune Interval	Foundry/configure/ip/pim#join-prune-interval <time#
Configure MRT Period	Foundry/configure/ip/pim#mrt-period <time#
Configure MRT Stale Multiplier	Foundry/configure/ip/pim#mrt-stale-mult <number#
Configure MRT SPT Multiplier	Foundry/configure/ip/pim#mrt-spt-multiplier <number#
Configure Probe Period	Foundry/configure/ip/pim#probe-period <time#
Configure Registration suppression timeout	Foundry/configure/ip/pim#register-suppress-timeout <time#
Configure DR to switch immediate	Foundry/configure/ip/pim#dr-switch-immediate
Configure RP to switch immediate	Foundry/configure/ip/pim#rp-switch-immediate
Configure Threshold for DR	Foundry/configure/ip/pim#threshold-dr <bps#
Configure Threshold for RP	Foundry/configure/ip/pim#threshold-rp <bps#
Configure to calculate whole packet checksum (for cisco interop)	Foundry/configure/ip/pim#whole-packet-checksum
Bootstrap Router related Commands	
Configure as candidate BSR	Foundry/configure/ip/pim/cbsr# address <address#
Configure CBSR period	Foundry/configure/ip/pim/cbsr# period <time#
Configure CBSR holdtime	Foundry/configure/ip/pim/cbsr#holdtime <time#
Configure CBSR priority	Foundry/configure/ip/pim/cbsr#priority <value#
RP commands	
Configure as candidate RP	Foundry/configure/ip/pim#crp
Configure as candidate RP address	Foundry/configure/ip/pim/crp# address <ipaddress#
Configure candidate RP group for advertisement	Foundry/configure/ip/pim/crp# group-add <address# [mask] [priority]
Configure as candidate RP holdtime	Foundry/configure/ip/pim/crp#holdtime <time#

TABLE 4 PIM COMMANDS (CONTINUED)

Configure as candidate RP period	Foundry/configure/ip/pim/crp#period <time#
Configure as candidate RP priority	Foundry/configure/ip/pim/crp#priority <value#
Configure a static RP address	Foundry/configure/ip/pim/# rp <address# <gaddress# [mask]
Interface based parameters	
Configure PIM for an interface	Foundry/configure/ip/pim#interface <interface_name#[:dlci_no]
Configure PIM mode for an interface	Foundry/configure/ip/pim/interface wan1# mode [sparse | dense | ssm | sparse-ssm]
Configure PIM interface assert holdtime	Foundry/configure/ip/pim/interface wan1#assert-holdtime <time#
Configure PIM interface hello holdtime	Foundry/configure/ip/pim/interface wan1#hello-holdtime <time#
Configure PIM interface hello interval	Foundry/configure/ip/pim/interface wan1#hello-interval <time#
Configure PIM interface Join/Prune Delay Timeout	Foundry/configure/ip/pim/interface wan1#join-prune-timeout <time#
Configure PIM interface Join/Prune Interval	Foundry/configure/ip/pim/interface wan1#join-prune-interval <time#
Configure PIM interface Join/Prune holdtime	Foundry/configure/ip/pim/interface wan1#join-prune-holdtime <time#
Configure PIM interface as border of PIM domain	Foundry/configure/ip/pim/interface wan1#boundary
SSM range	
Configure the SSM range	Foundry/configure/ip/pim# ssm-range <group-address# <group-mask

The show and debug PIM commands are:

TABLE 5 PIM SHOW AND DEBUG COMMANDS

Display PIM global configuration	Foundry#show ip pim global
Display PIMC timers	Foundry#show ip pim timers
Display PIM interfaces	Foundry#show ip pim interfaces
Display PIM neighbors	Foundry#show ip pim neighbors
Display PIM Bootstrap info	Foundry#show ip pim bsr-info

TABLE 5 PIM SHOW AND DEBUG COMMANDS (CONTINUED)

Display PIM Candidate RP info	Foundry#show ip pim crp-info
Display PIM statistics	Foundry#show ip pim statistics
Display PIM RP set	Foundry#show ip pim rp-set
Display PIM Static RP	Foundry#show ip pim rp
Trace PIM packets	Foundry# debug ip pim packet <pkt_type# <direction# [interface_name] [dlci]
Trace PIM state changes	Foundry# debug ip pim state
Trace PIM routes	Foundry# debug ip pim route
Trace PIM detail	Foundry# debug ip pim detail
Trace PIM debug	Foundry# debug ip pim debug
All Traces	Foundry#debug ip pim all

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.

The PIM-SSM command is 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:

- 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:

- 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:

TABLE 6 IGMP COMMANDS

Enabling igmp	Foundry/configure > ip igmp
Disabling igmp	Foundry/configure > no ip igmp
Enabling igmp	Foundry/configure/ip/igmp# interface <interface-name#[: dlci-no]
Disabling igmp	Foundry/configure/ip/igmp# no interface <interface-name#
Configuring version	Foundry/configure/ip/igmp/interface ethernet0# version <version#
Configuring Query Interval	Foundry/configure/ip/igmp/interface ethernet0# query-interval <interval#
Configuring Maximum Response Time	Foundry/configure/ip/igmp/interface ethernet0# query-response-interval <interval#
Configuring Last Member Query Interval	Foundry/configure/ip/igmp/interface ethernet0# last-member-query-interval <interval#
Configuring Last Member Query Count	Foundry/configure/ip/igmp/interface ethernet0# last-member-query-count <value#
Configuring Startup Query Interval	Foundry/configure/ip/igmp/interface ethernet0# startup-query-interval <interval#
Configuring Startup Query Count	Foundry/configure/ip/igmp/interface ethernet0# startup-query-count <count#
Configuring Robustness	Foundry/configure/ip/igmp/interface ethernet0# robustness <value#
Configuring Ignore-v1-message	Foundry/configure/ip/igmp/interface ethernet0# [no] ignore-v1-messages
Configuring Ignore-v2-message	Foundry/configure/ip/igmp/interface ethernet0# [no] ignore-v2-messages

TABLE 6 IGMP COMMANDS (CONTINUED)

Configuring Send Router Alerts	Foundry/configure/ip/igmp/interface ethernet0# [no] send-router-alert
Configuring Require Router Alerts	Foundry/configure/ip/igmp/interface ethernet0# [no] require-router-alert
Assigning filter list for group filtering	Foundry/configure/ip/igmp/interface ethernet0# group-filter <filter-list-name#
Debug Command	
Enable all debug levels	Foundry/debug#[no] ip igmp all
Debug state related events	Foundry/debug#[no] ip igmp state
Debug normal events	Foundry/debug#[no] ip igmp normal
Debug query packets	Foundry/debug#[no] ip igmp packet query [inbound | outbound]
Debug report packets	Foundry/debug#[no] ip igmp packet report [inbound | outbound]
Debug leave packets	Foundry/debug#[no] ip igmp packet leave [inbound | outbound]
Show Commands	
Displaying IGMP group membership information	Foundry# show ip igmp groups {all | <interface-name#} [detail]
Displaying IGMP interface configuration	Foundry# show ip igmp interface {all | <interface-name#}
Clear Command	
Clearing IGMP group membership information	Foundry# 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, but the path may still be traced.

Foundry supports the following PIM related feature—a “traceroute” facility for IP multicast, as defined in draft-ietf-idmr-traceroute-ipm-05.

The **mtrace** command for multicast traffic is similar to the **traceroute** command used for unicast traffic. Unlike **traceroute**, however, **mtrace** traces traffic backwards, from the receiver to the source. **mtrace** uses other unicast routing tables for RPF. For these, **mtrace** relies on Foundry’ implementation of the **mtrace** protocol is manageable through the CLI and can be executed from any command sub-tree of the Foundry CLI.

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, Foundry-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:

TABLE 7 MULTIPATH COMMANDS

Enabling HRW method	Foundry/configure/ip/multicast# multipath
Enabling Cisco method	Foundry/configure/ip/multicast# multipath cisco
Disabling Multipath	Foundry/configure/ip/multicast# no multipath Foundry/configure/ip/multicast# no multipath cisco
Display RPF selection	Foundry#show ip rpf <addr# <addr# - source or RP address

When multipath is disabled, Foundry 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)

Generic Routing Encapsulation (GRE) handles the transportation of IP multicast traffic between two sites that have only IP unicast connectivity. Using tunnels in a VPN environment is important because IPSec encryption is limited to IP unicast frames. IPSec over GRE tunneling allows for the encryption and the transportation of multiprotocol traffic across the VPN since both unicast and multicast IP packets appear to the IPSec protocol as IP unicast frame after GRE tunneling. If all connectivity must go through the home gateway router, tunnels also enable the use of private network addressing across a service provider's backbone without the need for running the Network Address Translation (NAT) feature.

The Foundry Network 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 *GRE Configuration Guide.*

Chapter 15
Security Features

Introduction to Security

Foundry introduces a wide range of robust industry-standard security features including:

- Virtual Private Networking
- IPSec encryption and tunneling
- Generalized Router Encryption
- Firewall with private network management (Network Address Translation and Port Address Translation)

This chapter explains each of these features in detail.

Enabling Security Features

The advanced VPN and firewall (`advance_vpn`) license allows users to manage remote LANs. This license also includes Basic VPN and Firewall licenses.

To see the license available in this release, enter:

```
Foundry/configure# system licenses ?

NAME
  licenses - Configure feature upgrade licenses

SYNTAX
  licenses license_type <cr#

DESCRIPTION
  license_type        -- Specifies the type of feature upgrade license
    The parameter may have any of the following values:
    advance_vpn       -- Enable Advance VPN and Firewall License
```

To install the advanced VPN and firewall license and use all the security features available in this release, enter:

```
Foundry/configure# system licenses advance_vpn
Enter Security Upgrade License key: 024f3bc296b4ea7265
```

Securing Remote Access Using IPSec VPN

The 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.

Access Methods

Foundry supports two types of IPSec remote access using VPNs.

Remote Access: User Group

One of the methods to achieve IPSec remote access in Foundry is the user group method. In this method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. Each user in the group is identified with unique information that is uniquely configured in the IKE policy. Also, an IPSec template is attached to the user group.

Once the VPN user is authenticated using IKE, the users dynamically-assigned IP address is added to the destination address field in the IPSec template attached to the user group. The VPN user now has the required IPSec policy that allows access through the gateway to the corporate LAN.

Remote Access: Mode Configuration

The other method to achieve IPSec remote access in Foundry is the mode configuration method.

This method makes the VPN client an extension of the LAN being accessed by the VPN client. The remote client appears as a network accessing some resource behind the VPN server.

The VPN client is allocated a private IP address by the VPN server and the client uses this as the source IP address in the inner IP header in tunnel mode.

In tunnel mode, at each IKE end point, the IP traffic to be protected is completely encapsulated with another IP packet. In this, the inner IP header remains the same as seen in the original traffic to be protected. In the outer IP header, the source and destination addresses are the addresses of the tunnel end points.

Typically, for a remote user, the source address of the outer IP header is the dynamic public IP address provided by the ISP. When mode configuration is enabled, the source address of the inner IP header is the private address allocated by the VPN server to the VPN client.

As in the case of user group method, the administrator creates an IKE policy for a logical group of users such as a department in an organization. The identity information used to identify each user uniquely is configured in the IKE policy. The IKE policy is attached to a mode configuration record. The mode configuration record contains an IPSec policy template to be used for creating dynamic IPSec policy. Also, the record contains one or more pools of private IP addresses to be used for allocating the addresses to the VPN clients. Besides the private IP address, the VPN server can also provide WINS and DNS server addresses.

Upon successful IKE authentication of a VPN client, the server checks whether the IKE policy used to authenticate the VPN client is enabled for mode configuration. If so, the server allocates a private IP address from one of the IP pools in the mode configuration record to the VPN client. The destination address field in the IPSec template attached to the user group is filled in with the private IP address allocated to the VPN client and this is installed as an IPSec policy.

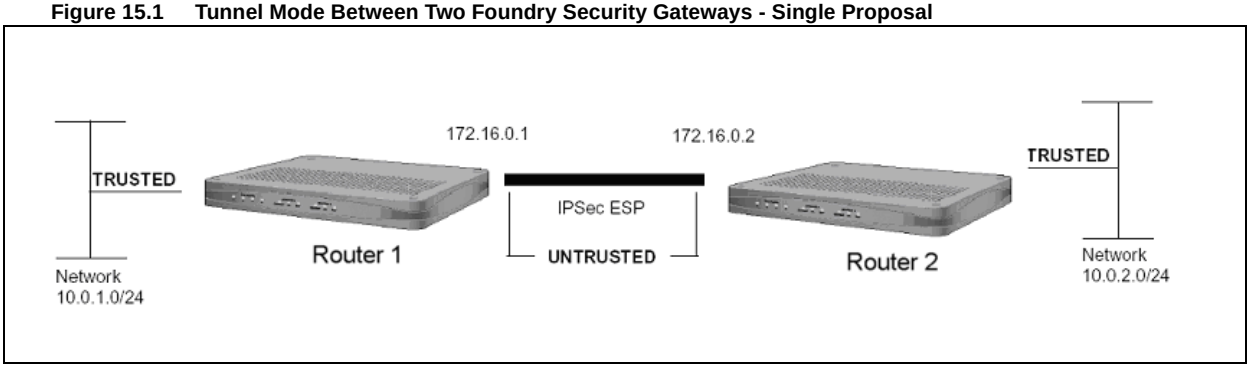
Example 1: Securely Managing the Foundry AR1204 Over an IPSec Tunnel

This example demonstrates how to manage a Foundry router through an IP security tunnel. Steps are presented for configuring the Router1 and Router2 routers to assist any host on the LAN side of Foundry-2 to manage the Router1 router through the IP security tunnel.

The security requirements are:

- Phase 1: 3DES with SHA1
- Phase 2: IPSec ESP with 128-bit AES and HMAC-SHA1

Figure 15.1 Tunnel Mode Between Two Foundry Security Gateways - Single Proposal



Step 1: Configure a WAN bundle of network type untrusted:

```
Router1/configure# interface bundle wan1
Configuring new bundle
Router1/configure/interface/bundle wan1# link t1 1
Router1/configure/interface/bundle wan1# encapsulation ppp
Router1/configure/interface/bundle wan1# ip address 172.16.0.1 24
Router1/configure/interface/bundle wan1# crypto untrusted
Router1/configure/interface/bundle wan1# exit
```


Step 2: Configure the Ethernet interface with trusted network type:

```
Router1/configure# interface ethernet 0
Configuring existing Ethernet interface
Router1/configure interface/ethernet 0# ip address 10.0.1.1 24
Router1/configure/interface/ethernet 0# crypto trusted
Router1/configure/interface/ethernet 0# exit
```

Step 3: Display the crypto interfaces:

```
Router1# show crypto interfaces

Interface       Network
Name            Type
---------       -------
ethernet0       trusted
wan1            untrusted
```

Step 4: Add the route to the peer LAN:

```
Router1/configure# ip route 10.0.2.0 24 wan1
```

Step 5: Configure IKE to the peer gateway:

```
Router1/configure# crypto
Router1/configure/crypto# ike policy Router2 172.16.0.2
Router1/configure/crypto/ike/policy Router2 172.16.0.2# local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-
pre_shared-g1
message: Key String has to be configured by the user
Router1/configure/crypto/ike/policy Router2 172.16.0.2# key
secretkey
Router1/configure/crypto/ike/policy Router2 172.16.0.2# proposal 1
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
encryption-al
algorithm 3des-cbc
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
exit
Router1/configure/crypto/ike/policy Router2 172.16.0.2# exit
```

Step 6: Display the IKE policies:

```
Router1# show crypto ike policy all

Policy     Peer            Mode       Transform
------     ----            ----       ---------
Router2    172.16.0.2      Main       P1 pre-g1-3des-sha1
```

Step 7: Display the IKE policies in detail:

```
Router1# show crypto ike policy all detail

Policy name Router2, Local addr 172.16.0.1, Peer addr 172.16.0.2
Main mode, Response and Initiate, PFS is not enabled, Shared Key is
*****
Local ident 172.16.0.1 (ip-address), Remote Ident 172.16.0.2 (ip-
address)

Proposal of priority 1
        Encryption algorithm: 3des
        Hash Algorithm: sha1
        Authentication Mode: pre-shared-key
        DH Group: group1
        Lifetime in seconds: 86400
        Lifetime in kilobytes: unlimited
```

Step 8: Configure the IPSec tunnel to the remote host:

```
Router1/configure/crypto# ipsec policy Router2 172.16.0.2
Router1/configure/crypto/ipsec policy Router2 172.16.0.2# match
address 172.16.0.1 32 10.0.2.0 24
   message: Default proposal created with
   priority1-esp-3des-sha1-tunnel and activated.

Router1/configure/crypto# ipsec policy Router2 172.16.0.2# proposal
1
Router1/configure/crypto# ipsec policy Router2 172.16.0.2/proposal
1# encryption-algorithm aes128-cbc
Router1/configure/crypto# ipsec policy Router2 172.16.0.2/proposal
1# exit
Router1/configure/crypto# ipsec policy Router2 172.16.0.2# exit
```

NOTE: For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix “IN” to the name.

Step 9: Display the IPSec policies:

```
Router1# show crypto ipsec policy all

Policy     Peer            Match                      Proto Transform
------     ----            -----                      ----- ---------
Router2    172.16.0.2      S 172.16.0.1/32/any        Any  P1 esp-aes-
sha1-tunl
                           D 10.0.2.0/24/any
INRouter2  172.16.0.2      S 10.0.2.0/24/any          Any  P1 esp-aes-
```

Step 10: Display IPSec policies in detail:

```
Router1# show crypto ipsec policy all detail

Policy name Router2 is enabled, Direction is outbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (172.16.0.1/255.255.255.255/
any)
        Destination ip address (ip/mask/port): (10.0.2.0/
255.255.255.0/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes128(key length=128 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000

Policy name INRouter2 is enabled, Direction is inbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Destination ip address (ip/mask/port): (172.16.0.1/
255.255.255.255/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes128(key length=128 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1000 in service ike self
Router1/configure/firewall internet/policy 1000 in# exit
Router1/configure/firewall internet# exit
```

Step 12: Configure firewall policies to allow desired services through untrusted interface to manage the router (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1001 in service snmp self
Router1/configure/firewall internet/policy 1001 in# exit
Router1/configure/firewall internet# policy 1002 in service telnet
self
Router1/configure/firewall internet/policy 1002 in# exit
Router1/configure/firewall internet# policy 1003 in protocol icmp
self
Router1/configure/firewall internet/policy 1003 in# exit
Router1/configure/firewall internet# exit
```

Step 13: Display firewall policies in the internet map (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
1000 in  any                any                ike               PERMIT SE
1001 in  any                any                snmp              PERMIT SE
1002 in  any                any                telnet            PERMIT SE
1003 in  any                any                any   any   icmp  PERMIT SE
1024 out any                any                any   any   any   PERMIT SE
```

Step 14: Display firewall policies in the internet map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is ike
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1001 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is snmp
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1002 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is telnet
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1003 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, Protocol is icmp
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
```

Step 15: Enable SNMP on the Router1 router:

```
Router1/configure/crypto/# exit
Router1/configure# snmp
Router1/configure/snmp# community public rw
Router1/configure/snmp# exit
```

Step 16: Display SNMP communities:

```
Router1# show snmp communities

Community = public, privilege=rw
```

Step 17: Repeat steps 1 - 16 with suitable modifications on Router2 prior to managing Router1 from Router2's LAN side.

Step 18: Test the IPSec tunnel for managing the Router1 router from a host on Router2's LAN.

Step 19: When the SNMP manager starts managing Router1 from Router2's LAN, display the IKE and IPSec SA tables.

```
Router1# show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------
Router2    172.16.0.2      SA_MATURE  2020       pre-g1-3des-sha1
```

```
Router1# show crypto ike sa all detail

Crypto Policy name: Router2
        Remote  ident 172.16.0.2
        Peer Address is 172.16.0.2
        Transform: 3des, sha1, pre-shared-key
        DH Group: group1
        Bytes Processed 2020
        State is SA_MATURE
        Mode is Main
        Remaining Time in Sec: 86084
        Life Time in Sec: 86400, Life Time in Bytes is unlimited
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INRouter2  172.16.0.1      0xe8453c2b 256        esp-aes-sha1-tunl
Router2    172.16.0.2      0xa1f673aa 256        esp-aes-sha1-tunl
```

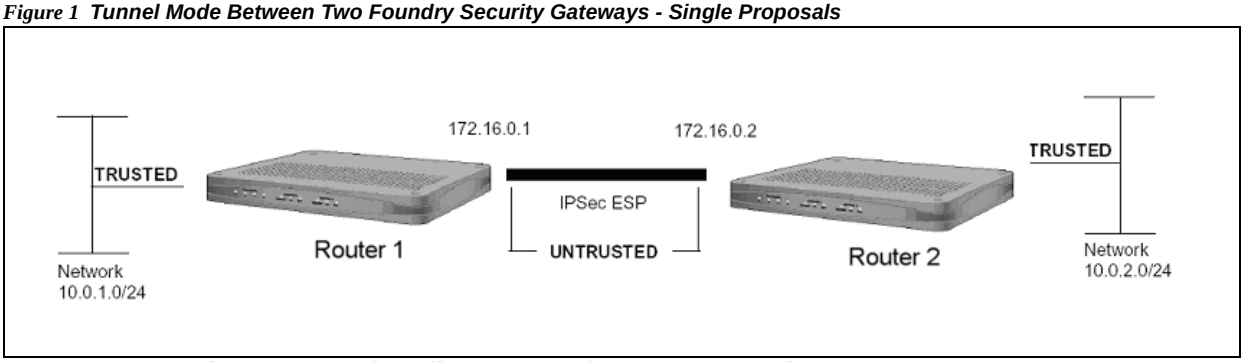
```
Router1# show crypto ipsec sa all detail

Crypto Policy name: INRouter2
        Protocol is Any
        Local  ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Remote ident(ip/mask/port): (172.16.0.1/255.255.255.255/any)
        Peer Address is 172.16.0.1, PFS Group is disabled

inbound ESP sas
        Spi: 0xe8453c2b
        Transform: aes128 (key length=128 bits), sha1
        In use settings = {tunnel}
        Bytes Processed 256
        Hard lifetime in seconds 3290, Hard lifetime in kilobytes
413696
        Soft lifetime in seconds 0, Soft lifetime in kilobytes is
unlimited

Crypto Policy name: Router2
        Protocol is Any
        Local  ident(ip/mask/port): (172.16.0.1/255.255.255.255/any)
        Remote ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Peer Address is 172.16.0.2, PFS Group is disabled

outbound ESP sas
        Spi: 0xa1f673aa
        Transform: aes128 (key length=128 bits), sha1
```

Example 2: Joining Two Private Networks with an IP Security Tunnel

The following example demonstrates how to form an IP security tunnel to join two private networks: 10.0.1.0/24 and 10.0.2.0/24. The security requirements are as follows:

- Phase 1: 3DES with SHA1
- Phase 2: IPSec ESP with AES (256-bit) and HMAC-SHA1

Figure 1 ***Tunnel Mode Between Two Foundry Security Gateways - Single Proposals***



Step 1: Configure a WAN bundle of network type untrusted:

```
Router1/configure/interface/bundle wan1# link t1 1
Router1/configure/interface/bundle wan1# encapsulation ppp
Router1/configure/interface/bundle wan1# ip address 172.16.0.1 24
Router1/configure/interface/bundle wan1# crypto untrusted
Router1/configure/interface/bundle wan1# exit
```

Step 2: Configure the Ethernet interface with trusted network type:

```
Router1/configure# interface ethernet 0
Configuring existing Ethernet interface
Router1/configure interface/ethernet 0# ip address 10.0.1.1 24
Router1/configure/interface/ethernet 0# crypto trusted
Router1/configure/interface/ethernet 0# exit
```

Step 3: Display the crypto interfaces:

```
Router1# show crypto interfaces

Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Add route to peer LAN:

```
Router1/configure# ip route 10.0.2.0 24 wan1
```

Step 5: Configure IKE to the peer gateway:

```
Router1/configure# crypto
Router1/configure/crypto# ike policy Router2 172.16.0.2
Router1/configure/crypto/ike/policy Router2 172.16.0.2# local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-
pre_shared-g1
message: Key String has to be configured by the user
Router1/configure/crypto/ike/policy Router2 172.16.0.2# key
secretkey
Router1/configure/crypto/ike/policy Router2 172.16.0.2# proposal 1
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
encryption-al
algorithm 3des-cbc
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
exit
Router1/configure/crypto/ike/policy Router2 172.16.0.2# exit
```

Step 6: Display the IKE policies:

```
Router1# show crypto ike policy all

Policy     Peer            Mode       Transform
------     ----            ----       ---------
Router2    172.16.0.2      Main       P1 pre-g1-3des-sha1
```

Step 7: Display the IKE policies in detail:

```
Router1# show crypto ike policy all detail

Policy name Router2, Local addr 172.16.0.1, Peer addr 172.16.0.2
Main mode, Response and Initiate, PFS is not enabled, Shared Key is
*****
Local ident 172.16.0.1 (ip-address), Remote Ident 172.16.0.2 (ip-
address)

Proposal of priority 1
        Encryption algorithm: 3des
        Hash Algorithm: sha1
        Authentication Mode: pre-shared-key
        DH Group: group1
        Lifetime in seconds: 86400
        Lifetime in kilobytes: unlimited
```

Step 8: Configure IPSec tunnel to the remote host:

```
Router1/configure/crypto# ipsec policy Router2 172.16.0.2
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# match
address 10.0.1.0 24 10.0.2.0 24
Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# proposal 1
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 1#
encryption-algorithm aes256-cbc
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 1#
exit
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# exit
```

NOTE: For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix “IN” to the name.

Step 9: Display IPSec policies:

```
Router1# show crypto ipsec policy all

Policy     Peer            Match                     Proto Transform
------     ----            -----                     ----- ---------
Router2    172.16.0.2      S 10.0.1.0/24/any          Any  P1 esp-aes-
sha1-tunl
```

Step 10: Display IPSec policies detail:

```
Router1# show crypto ipsec policy all detail

Policy name Router2 is enabled, Direction is outbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.1.0/255.255.255.0/any)
        Destination ip address (ip/mask/port): (10.0.2.0/
255.255.255.0/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000

Policy name INRouter2 is enabled, Direction is inbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Destination ip address (ip/mask/port): (10.0.1.0/
255.255.255.0/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1000 in service ike self
Router1/configure/firewall internet/policy 1000 in# exit
Router1/configure/firewall internet# exit
```

Step 12: Display firewall policies in the internet map (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
1000 in  any                any                ike               PERMIT SE
1024 out any                any                any   any   any   PERMIT SE
```

Step 13: Display firewall policies in the internet map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is ike
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
```

Step 14: Configure firewall policies to allow transit traffic from remote LAN to the local LAN (applicable only if firewall license is also enabled):

```
Router1/configure# firewall corp
Router1/configure/firewall corp# policy 1000 in address 10.0.2.0 24
10.0.1.0 24
Router1/configure/firewall corp/policy 1000 in# exit
Router1/configure/firewall corp# exit
```

Step 15: Display firewall policies in the corp map (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto
Action Advanced
---  --- -----------        ----------------   ----------------- ----
-- --------
```

Step 16: Display firewall policies in the corp map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is transit
Logging is disable
Source Address is 10.0.2.0/24, Dest Address is 10.0.1.0/24
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1022 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1023 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is transit
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
```

Step17: Repeat steps 1 -16 with suitable modifications on Router2 prior to passing traffic.

Step 18: Test the IPSec tunnel between Router1 and Router2 by passing traffic from the 10.0.1.0 to the 10.0.2.0 network.

Step 19: After transit traffic is passed through the tunnel, display the IKE and IPSec SA tables.

```
Router1# show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------
Router2    172.16.0.2      SA_MATURE  1796       pre-g1-3des-sha1
```

```
Router1# show crypto ike sa all detail

Crypto Policy name: Router2
        Remote  ident 172.16.0.2
        Peer Address is 172.16.0.2
        Transform: 3des, sha1, pre-shared-key
        DH Group: group1
        Bytes Processed 1796
        State is SA_MATURE
        Mode is Main
        Remaining Time in Sec: 86376
        Life Time in Sec: 86400, Life Time in Bytes is unlimited
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INRouter2  172.16.0.1      0xd603a513 256        esp-aes-sha1-tunl
Router2    172.16.0.2      0xb013de87 256        esp-aes-sha1-tunl
```

```
Router1# show crypto ipsec sa all detail

Crypto Policy name: INRouter2
        Protocol is Any
        Local  ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Remote ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
        Peer Address is 172.16.0.1, PFS Group is disabled

inbound ESP sas
        Spi: 0xd603a513
        Transform: aes256 (key length=256 bits), sha1
        In use settings = {tunnel}
        Bytes Processed 256
        Hard lifetime in seconds 3560, Hard lifetime in kilobytes
413696
        Soft lifetime in seconds 0, Soft lifetime in kilobytes is
unlimited

Crypto Policy name: Router2
        Protocol is Any
        Local  ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
        Remote ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Peer Address is 172.16.0.2, PFS Group is disabled

outbound ESP sas
        Spi: 0xb013de87
        Transform: aes256 (key length=256 bits), sha1
```

Example 3: Joining Two Networks with an IPSec Tunnel using Multiple IPSec Proposals

The following example demonstrates how a security gateway can use multiple IPSec (phase2) proposals to form an IP security tunnel to join two private networks: 10.0.1.0/24 and 10.0.2.0/24.

IKE Proposal offered by both Router1 and Router2:

- Phase 1: 3DES and SHA1

IPSec Proposals offered by Router1:

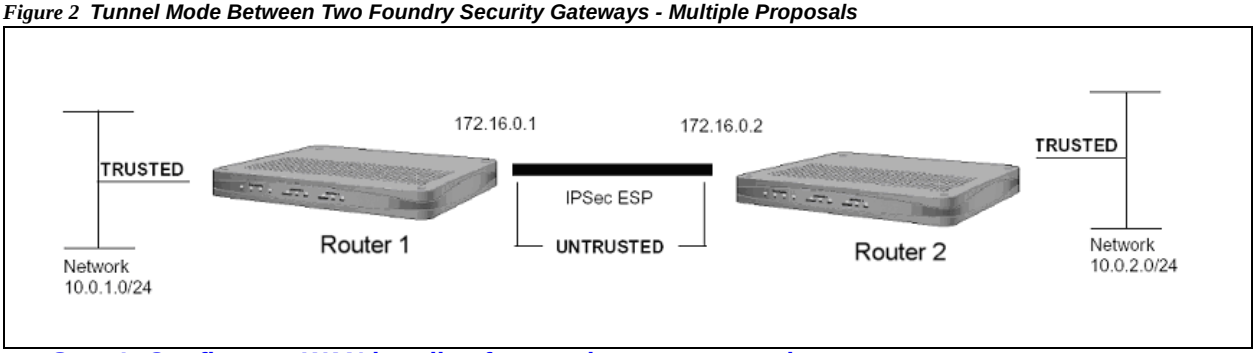
- Phase 2: Proposal1: IPSec ESP with DES and HMAC-SHA1
- Phase 2: Proposal2: IPSec ESP with AES (256-bit) and HMAC-SHA1

IPSec Proposal offered by Router2:

- Phase 2: Proposal1: IPSec ESP with AES (256-bit) and HMAC-SHA1

In this example, the Router1 router offers two IPSec proposals to the peer while the Router2 router offers only one proposal. As a result of quick mode negotiation, the two routers are expected to converge on a mutually acceptable proposal, which is the proposal “IPSec ESP with AES (256-bit) and HMAC-SHA1” in this example.

Figure 2 ***Tunnel Mode Between Two Foundry Security Gateways - Multiple Proposals***



Step 1: Configure a WAN bundle of network type untrusted:

```
Router1/configure/interface/bundle wan1# link t1 1
Router1/configure/interface/bundle wan1# encapsulation ppp
Router1/configure/interface/bundle wan1# ip address 172.16.0.1 24
Router1/configure/interface/bundle wan1# crypto untrusted
Router1/configure/interface/bundle wan1# exit
```

Step 2: Configure the Ethernet interface with trusted network type:

```
Router1/configure# interface ethernet 0
Configuring existing Ethernet interface
Router1/configure interface/ethernet 0# ip address 10.0.1.1 24
Router1/configure/interface/ethernet 0# crypto trusted
Router1/configure/interface/ethernet 0# exit
```

Step 3: Display the crypto interfaces:

```
Router1# show crypto interfaces

Interface       Network
Name            Type
---------       -------
ethernet0       trusted
wan1            untrusted
```

Step 4: Add the route to the peer LAN:

```
Router1/configure# ip route 10.0.2.0 24 wan1
```

Step 5: Configure IKE to the peer gateway:

```
Router1/configure# crypto
Router1/configure/crypto# ike policy Router2 172.16.0.2
Router1/configure/crypto/ike/policy Router2 172.16.0.2# local-
address 172.16.0.1
message: Default proposal created with priority1-des-sha1-
pre_shared-g1
message: Key String has to be configured by the user
Router1/configure/crypto/ike/policy Router2 172.16.0.2# key
secretkey
Router1/configure/crypto/ike/policy Router2 172.16.0.2# proposal 1
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
encryption-al
gorithm 3des-cbc
Router1/configure/crypto/ike/policy Router2 172.16.0.2/proposal 1#
exit
Router1/configure/crypto/ike/policy Router2 172.16.0.2# exit
```

Step 6: Display the IKE policies:

```
Router1# show crypto ike policy all

Policy     Peer            Mode       Transform
------     ----            ----       ---------
Router2    172.16.0.2      Main       P1 pre-g1-3des-sha1
```

Step 7: Display the IKE policies in detail:

```
Router1# show crypto ike policy all detail

Policy name Router2, Local addr 172.16.0.1, Peer addr 172.16.0.2
Main mode, Response and Initiate, PFS is not enabled, Shared Key is
*****
Local ident 172.16.0.1 (ip-address), Remote Ident 172.16.0.2 (ip-
address)

Proposal of priority 1
        Encryption algorithm: 3des
        Hash Algorithm: sha1
        Authentication Mode: pre-shared-key
        DH Group: group1
        Lifetime in seconds: 86400
        Lifetime in kilobytes: unlimited
```

Step 8: Configure IPSec tunnel to the remote host:

```
Router1/configure/crypto# ipsec policy Router2 172.16.0.2
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# match
address 10.0.1.0 24 10.0.2.0 24
Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# proposal 1
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 1#
encryption-algorithm des-cbc
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 1#
exit
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# proposal 2
Proposal added with priority2-esp-3des-sha1-tunnel.
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 2#
encryption-algorithm aes256-cbc
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2/proposal 2#
exit
Router1/configure/crypto/ipsec/policy Router2 172.16.0.2# exit
Router1/configure/crypto# exit
Router1/configure#
```

NOTE: For IPSec only – when you create an outbound tunnel, an inbound tunnel is automatically created. The inbound tunnel applies the name that you provide for the outbound tunnel and adds the prefix “IN” to the name.

Step 9: Display the IPSec policies:

```
Router1# show crypto ipsec policy all

Policy     Peer            Match                    Proto Transform
------     ----            -----                    ----- ---------
Router2    172.16.0.2      S 10.0.1.0/24/any        Any  P1 esp-des-
sha1-tunl
```

```
Router1# show crypto ipsec policy all detail

Policy name Router2 is enabled, Direction is outbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.1.0/255.255.255.0/
any)
        Destination ip address (ip/mask/port): (10.0.2.0/
255.255.255.0/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: des
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000

Proposal of priority 2
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000

Policy name INRouter2 is enabled, Direction is inbound
Peer Address is 172.16.0.2, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.2.0/255.255.255.0/
any)
```

Step 10: Configure firewall policies to allow IKE negotiation through untrusted interface (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1000 in service ike self
Router1/configure/firewall internet/policy 1000 in# exit
Router1/configure/firewall internet# exit
```

Step 11: Display firewall policies in the internet map (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
1000 in  any                any                ike               PERMIT SE
1024 out any                any                any   any   any   PERMIT SE
```

Step 12: Display firewall policies in the internet map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is ike
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
```

Step 13: Configure firewall policies to allow transit traffic from remote LAN to the local LAN (applicable only if firewall license is also enabled):

```
Router1/configure# firewall corp
Router1/configure/firewall corp# policy 1000 in address 10.0.2.0 24
10.0.1.0 24
Router1/configure/firewall corp/policy 1000 in# exit
Router1/configure/firewall corp# exit
```

Step 14: Display firewall policies in the corp map (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto
Action Advanced
---  --- -----------        ----------------   ----------------- ----
-- --------
```


Step 15: Display firewall policies in the corp map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is transit
Logging is disable
Source Address is 10.0.2.0/24, Dest Address is 10.0.1.0/24
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1022 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1023 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is transit
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 11258, Bytes Out 5813
```

Step16: Repeat steps 1 -15 with suitable modifications on Router2 prior to passing bi-directional traffic.

Step 17: Test the IPSec tunnel between Router1 and Router2 by passing traffic from the 10.0.1.0 network to the 10.0.2.0 network.

Step 18: After traffic is passed through the tunnel, display the IKE and IPSec SA tables.

```
Router1# show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------
Router2    172.16.0.2      SA_MATURE  1796       pre-g1-3des-sha1
```

```
Router1# show crypto ike sa all detail

Crypto Policy name: Router2
        Remote  ident 172.16.0.2
        Peer Address is 172.16.0.2
        Transform: 3des, sha1, pre-shared-key
        DH Group: group1
        Bytes Processed 1796
        State is SA_MATURE
        Mode is Main
        Remaining Time in Sec: 86380
        Life Time in Sec: 86400, Life Time in Bytes is unlimited
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INRouter2  172.16.0.1      0x8eabe4b3 256        esp-aes-sha1-tunl
Router2    172.16.0.2      0xa9a506f9 256        esp-aes-sha1-tunl
```

```
Router1# show crypto ipsec sa all detail

Crypto Policy name: INRouter2
        Protocol is Any
        Local  ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Remote ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
        Peer Address is 172.16.0.1, PFS Group is disabled

inbound ESP sas
        Spi: 0x8eabe4b3
        Transform: aes256 (key length=256 bits), sha1
        In use settings = {tunnel}
        Bytes Processed 256
        Hard lifetime in seconds 3570, Hard lifetime in kilobytes
413696
        Soft lifetime in seconds 0, Soft lifetime in kilobytes is
unlimited

Crypto Policy name: Router2
        Protocol is Any
        Local  ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
        Remote ident(ip/mask/port): (10.0.2.0/255.255.255.0/any)
        Peer Address is 172.16.0.2, PFS Group is disabled

outbound ESP sas
        Spi: 0xa9a506f9
```

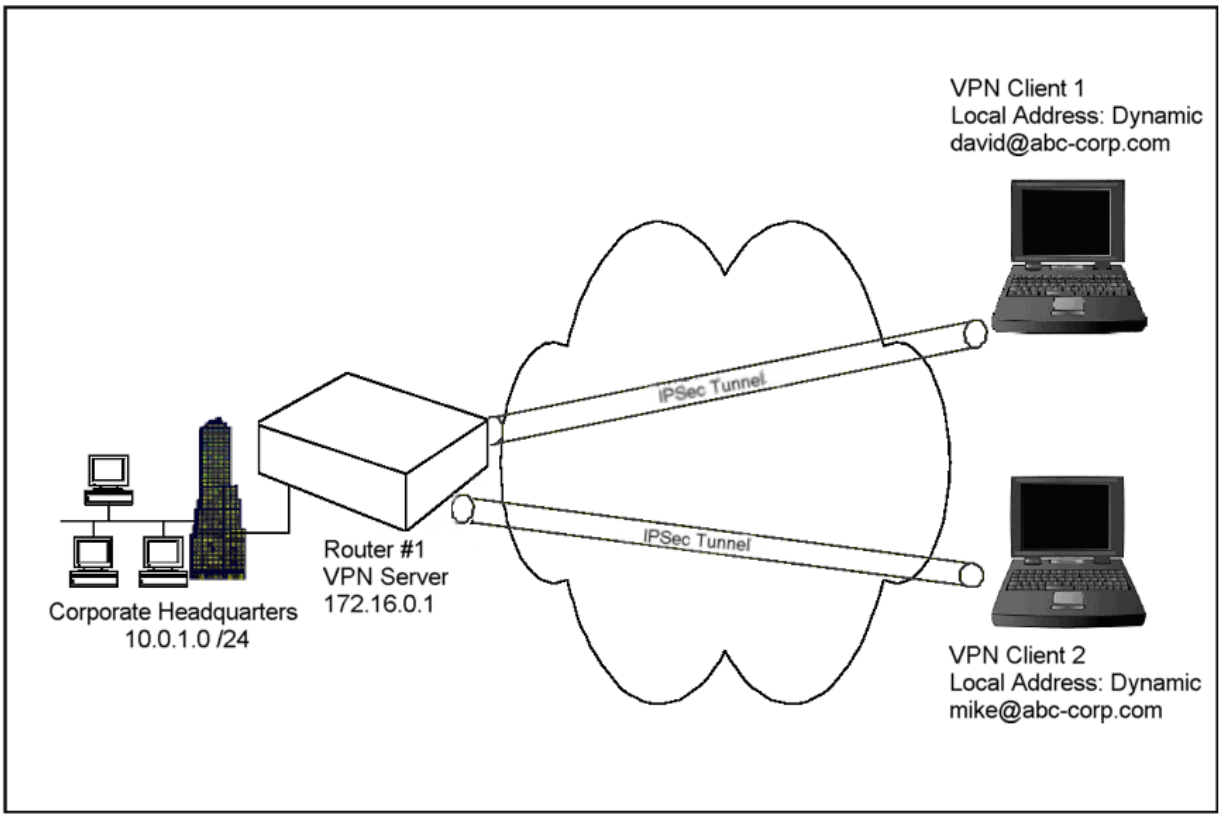
Example 4: Supporting Remote User Access

The following example demonstrates how to configure a Foundry router to be an IPSec VPN server using user group method with extended authentication (XAUTH) for remote VPN clients. The client could be any standard IPSec VPN client.

In this example, the client needs to access the corporate private network 10.0.1.0/24 through the VPN tunnel. The security requirements are as follows:

- Phase 1: 3DES with SHA1, Xauth (Radius PAP)
- Phase 2: IPSec ESP tunnel with AES256 and HMAC-SHA1

Figure 15.2 IPSec Tunneling Using User Group Method



Step 1: Configure a WAN bundle of network type untrusted:

```
Router1/configure# interface bundle wan1
Configuring new bundle
Router1/configure/interface/bundle wan1# link t1 1
Router1/configure/interface/bundle wan1# encapsulation ppp
Router1/configure/interface/bundle wan1# ip address 172.16.0.1 24
Router1/configure/interface/bundle wan1# crypto untrusted
Router1/configure/interface/bundle wan1# exit
```

Step 2: Configure the Ethernet interface with trusted network type:

```
Router1/configure# interface ethernet 0
Configuring existing Ethernet interface
Router1/configure interface/ethernet 0# ip address 10.0.1.1 24
Router1/configure/interface/ethernet 0# crypto trusted
Router1/configure/interface/ethernet 0# exit
```

Step 3: Display the crypto interfaces:

```
Router1# show crypto interfaces

Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Configure dynamic IKE policy for a group of mobile users:

```
Router1/configure# crypto
Router1/configure/crypto# dynamic
Router1/configure/crypto/dynamic# ike policy sales
Router1/configure/crypto/dynamic/ike/policy sales# local-address
172.16.0.1
Router1/configure/crypto/dynamic/ike/policy sales# remote-id email-id
david@abc-corp.com david

New user david is added to the group sales
Default proposal created with priority1-des-sha1-pre_shared-g1
Key String has to be configured by the user

Router1/configure/crypto/dynamic/ike/policy sales# remote-id email-id
mike@abc-corp.com mike
New user mike is added to the group sales

Router1/configure/crypto/dynamic/ike/policy sales# key
secretkeyforsalesusers
Router1/configure/crypto/dynamic/ike/policy sales# proposal 1
Router1/configure/crypto/dynamic/ike/policy sales/proposal 1#
encryption-algorithm 3des-cbc
Router1/configure/crypto/dynamic/ike/policy sales/proposal 1# exit
Router1/configure/crypto/dynamic/ike/policy sales# client
authentication radius pap
Router1/configure/crypto/dynamic/ike/policy sales# exit
Router1/configure/crypto/dynamic#
```

Step 5: Display dynamic IKE policies:

```
Router1# show crypto dynamic ike policy all

Policy     Remote-id       Mode       Transform           Address-Pool
------     ---------       ----       ---------           ------------
sales      U david@foun... Aggressive P1                  pre-g1-3des-
```

Step 6: Display dynamic IKE policies in detail:

```
Router1# show crypto dynamic ike policy all detail

Policy name sales, User group name sales
Aggressive mode, Response Only, PFS is not enabled, Shared Key is
*****
Client authentication is Radius(PAP)
Local addr: 172.16.0.1, Local ident 172.16.0.1 (ip-address)
Remote idents are david@abc-corp.com (email-id), mike@abc-corp.com
(email-id)

Proposal of priority 1
        Encryption algorithm: 3des
        Hash Algorithm: sha1
        Authentication Mode: pre-shared-key
        DH Group: group1
        Lifetime in seconds: 86400
        Lifetime in kilobytes: unlimited
```

Step 7: Configure dynamic IPSec policy for a group of mobile users:

```
Router1/configure/crypto/dynamic# ipsec policy sales

Router1/configure/crypto/dynamic/ipsec/policy sales# match address
10.0.1.0 24
Default proposal created with priority1-esp-3des-sha1-tunnel and
activated.

Router1/configure/crypto/dynamic/ipsec/policy sales# proposal 1
Router1/configure/crypto/dynamic/ipsec/policy sales/proposal 1#
encryption-algorithm aes256-cbc
Router1/configure/crypto/dynamic/ipsec/policy sales/proposal 1# exit
Router1/configure/crypto/dynamic/ipsec/policy sales# exit
Router1/configure/crypto/dynamic#
```

Step 8: Display dynamic IPSec policies:

```
Router1# show crypto dynamic ipsec policy all

Policy     Match                     Proto Transform
------     -----                     ----- ---------
sales      S 10.0.1.0/24/any          Any  P1 esp-aes-sha1-tunl
           D any/any/any
INsales    S any/any/any              Any  P1 esp-aes-sha1-tunl
           D 10.0.1.0/24/any
```

Step 9: Display dynamic IPSec policies in detail:

```
Router1# show crypto dynamic ipsec policy all detail

Policy sales is enabled, User group name sales
Direction is outbound, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.1.0/255.255.255.0/
any)
        Destination ip address (ip/mask/port): (any/any/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000


Policy INsales is enabled, User group name sales
Direction is inbound, Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (any/any/any)
        Destination ip address (ip/mask/port): (10.0.1.0/
255.255.255.0/any)

Proposal of priority 1
        Protocol: esp
        Mode: tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000
```

Step 10: Configure radius server (applicable only if client authentication is configured in dynamic IKE policy):

```
Router1/configure# aaa
Router1/configure/aaa# radius
Router1/configure/aaa/radius# primary_server 172.168.2.1
Primary Radius server configured.
Router1/configure/aaa/radius# secondary_server 192.168.2.1
Secondary Radius server configured.
Router1/configure/aaa/radius# exit
Router1/configure/aaa# exit
```

Step 11: Configure firewall policies to allow IKE negotiation through untrusted interface (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1000 in service ike self
Router1/configure/firewall internet/policy 1000 in# exit
Router1/configure/firewall internet# exit
```

Step 12: Display firewall policies in the internet map (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr       Destination Addr   Sport Dport Proto Action
Advanced
```

Step 13: Display firewall policies in the internet map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is ike
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0
```

Step 14: Configure firewall policies for a group of mobile users to allow access to the local LAN (applicable only if firewall license is enabled):

```
Router1/configure/firewall corp#
Router1/configure/firewall corp# policy 1000 in user-group sales
address any  any 10.0.1.0 24
Router1/configure/firewall corp/policy 1000 in# exit
Router1/configure/firewall corp#
```

NOTE: Be sure to match the user group name in the policy command with the name used in Step 4 (the dynamic IKE policy).

Step 15: Display firewall policies in the corp map (applicable only if firewall license is enabled).

```
Router1# show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto
Action Advanced
---  --- -----------        ----------------   ----------------- ----
```

Step 16: Display firewall policies in the corp map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is transit
User Group is sales, Logging is disable
Source Address is any, Dest Address is 10.0.1.0/24
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1022 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1023 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is transit
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 11258, Bytes Out 5813
```

Step 17: Test the IPSec tunnel between the VPN client and the server by passing traffic from the client to the 10.0.1.0 network.

Step 18: After passing traffic through the tunnel, display the list of clients logged onto the VPN server and the IKE and IPSec SA tables:

```
Router1# show crypto dynamic clients
Client Address    Client Id                Policy            Advanced
--------------    ---------                ------            --------
192.168.107.105 david@abc-corp...           sales             UserGrp
```

```
Router1# show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------
```

```
Router1# show crypto ike sa all detail

Crypto Policy name: sales
    Remote  ident david@abc-corp.com
    Peer Address is 192.168.107.105
    Transform: 3des, sha1, pre-shared-key
    DH Group: group1
    Bytes Processed 1772
    State is SA_MATURE
    Mode is Aggressive
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INsales    172.16.0.1  0xf43c5e3b 360        esp-aes-sha1-tunl
sales      192.168.107.105 0xcfea8435 240        esp-aes-sha1-tunl
```

```
T Router1# show crypto ipsec sa all detail

Crypto Policy name: INsales
    Protocol is Any
    Local  ident(ip/mask/port): (192.168.107.105/255.255.255.255/any)
    Remote ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
    Peer Address is 172.16.0.1, PFS Group is disabled

inbound ESP sas
    Spi: 0xf43c5e3b
    Transform: aes256 (key length=256 bits), sha1
    In use settings = {tunnel}
    Bytes Processed 360
    Hard lifetime in seconds 28780, Hard lifetime in kilobytes is
unlimited
    Soft lifetime in seconds 0, Soft lifetime in kilobytes is
unlimited

Crypto Policy name: sales
    Protocol is Any
    Local  ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
    Remote ident(ip/mask/port): (192.168.107.105/255.255.255.255/any)
```

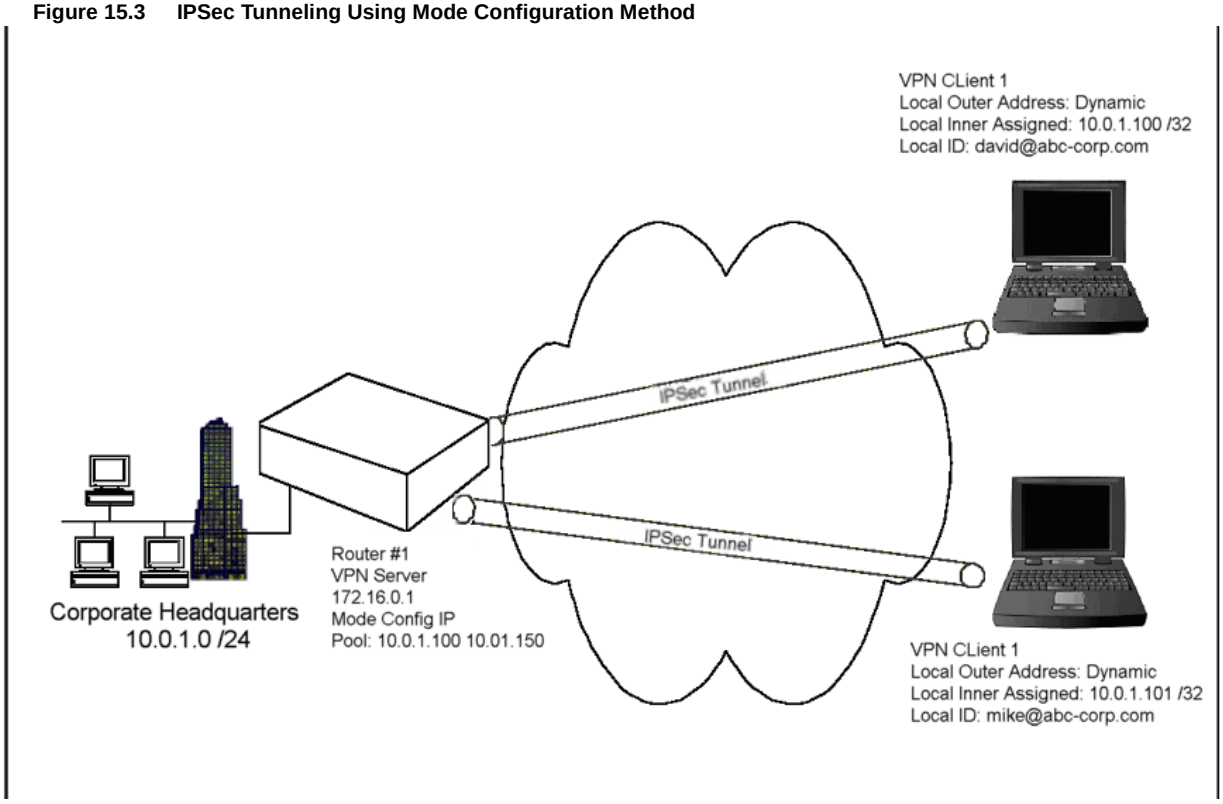
Example 5: Configuring IPSec Remote Access to Corporate LAN with Mode-Configuration Method

The following example demonstrates how to configure a Foundry router to be an IPSec VPN server using mode-configuration method. The client could be any standard mode configuration enabled IPSec VPN client.

In this example, the client needs to access the corporate private network 10.0.1.0/24 through the VPN tunnel. The server has a pool of ip addresses from 20.1.1.100 through 20.1.1.150 to be allocated for mode configuration enabled VPN clients. The assigned IP address will be used by the VPN client as the source address in the inner IP header. The outer IP header will carry the dynamic IP address assigned by the Internet Service Provider as the source address. The security requirements are as follows:

- Phase 1: 3DES with SHA1, Mode Configuration
- Phase 2: IPSec ESP tunnel with AES256 and HMAC-SHA1

Figure 15.3 IPSec Tunneling Using Mode Configuration Method



Step 1: Configure a WAN bundle of network type untrusted:

```
Router1/configure# interface bundle wan1
Configuring new bundle
Router1/configure/interface/bundle wan1# link t1 1
Router1/configure/interface/bundle wan1# encapsulation ppp
Router1/configure/interface/bundle wan1# ip address 172.16.0.1 24
Router1/configure/interface/bundle wan1# crypto untrusted
Router1/configure/interface/bundle wan1# exit
```

Step 2: Configure the Ethernet interface with trusted network type:

```
Router1/configure# interface ethernet 0
Configuring existing Ethernet interface
Router1/configure interface/ethernet 0# ip address 10.0.1.1 24
Router1/configure/interface/ethernet 0# crypto trusted
Router1/configure/interface/ethernet 0# exit
```

Step 3: Display the crypto interfaces:

```
Router1# show crypto interfaces

Interface      Network
Name           Type
---------      -------
ethernet0      trusted
wan1           untrusted
```

Step 4: Configure dynamic IKE policy for a group of mobile users:

```
Router1/configure# crypto
Router1/configure/crypto# dynamic
Router1/configure/crypto/dynamic# ike policy sales modecfg-group
Router1/configure/crypto/dynamic/ike/policy sales# local-address
192.168.55.52

Router1/configure/crypto/dynamic/ike/policy sales# remote-id email
david@abc-corp.com

Default proposal created with priority1-des-sha1-pre_shared-g1
Key String has to be configured by the user
Default ipsec proposal 'sales' added with priority1-3des-sha1-tunnel

Router1/configure/crypto/dynamic/ike/policy sales# remote-id email
mike@abc-corp.com

Router1/configure/crypto/dynamic/ike/policy sales# key
secretkeyforsales
Router1/configure/crypto/dynamic/ike/policy sales# proposal 1
Router1/configure/crypto/dynamic/ike/policy sales/proposal 1#
encryption-algorithm 3des-cbc
Router1/configure/crypto/dynamic/ike/policy sales/proposal 1# exit
Router1/configure/crypto/dynamic/ike/policy sales# client
configuration
Router1/configure/crypto/dynamic/ike/policy sales/client/
configuration# address-pool 1 20.1.1.100 20.1.1.150
Router1/configure/crypto/dynamic/ike/policy sales/client/
configuration# exit
Router1/configure/crypto/dynamic/ike/policy sales# exit
Router1/configure/crypto/dynamic# exit
```

Step 5: Display dynamic IKE policies:

```
Router1# show crypto dynamic ike policy all

Policy     Remote-id       Mode       Transform           Address-Pool
------     ---------       ----       ---------           ------------
sales       U david@foun... Aggressive P1 pre-g1-3des-sha1  1 S
```

Step 6: Display dynamic IKE policies in detail:

```
Router1# show crypto dynamic ike policy all detail

Policy name sales, Modeconfig group
Aggressive mode, Response Only, PFS is not enabled, Shared Key is
*****
Local addr: 192.168.55.52, Local ident 192.168.55.52 (ip-address)
Remote idents are david@abc-corp.com (email-id), mike@abc-corp.com
(email-id)
Address Pool:
        Pool# 1: 20.1.1.100 to 20.1.1.150

Proposal of priority 1
        Encryption algorithm: 3des
        Hash Algorithm: sha1
        Authentication Mode: pre-shared-key
        DH Group: group1
        Lifetime in seconds: 86400
        Lifetime in kilobytes: unlimited
```

Step 7: Configure dynamic IPSec policy for a group of mobile users:

```
Router1/configure/crypto#
Router1/configure/crypto# dynamic
Router1/configure/crypto/dynamic# ipsec policy sales modecfg-group
Router1/configure/crypto/dynamic/ipsec/policy sales# match address
10.0.1.0 24
Router1/configure/crypto/dynamic/ipsec/policy sales# proposal 1
Router1/configure/crypto/dynamic/ipsec/policy sales/proposal 1#
encryption-algorithm aes256-cbc
Router1/configure/crypto/dynamic/ipsec/policy sales/proposal 1# exit
Router1/configure/crypto/dynamic/ipsec/policy sales# exit
Router1/configure/crypto/dynamic# exit
```

Step 8: Display dynamic IPSec policies:

```
Router1# show crypto dynamic ipsec policy all

Policy     Match                     Proto Transform
------     -----                     ----- ---------
sales      S 10.0.1.0/24/any          Any  P1 esp-aes-sha1-tunl
           D any/any/any
```

Step 9: Display dynamic IPSec policies in detai:l

```
Router1# show crypto dynamic ipsec policy all detail

Policy sales is enabled, Modeconfig Group
Action is Apply
Key Management is Automatic
PFS Group is disabled
Match Address:
        Protocol is Any
        Source ip address (ip/mask/port): (10.0.1.0/255.255.255.0/
any)
        Destination ip address (ip/mask/port): (any/any/any)

Proposal of priority 1
        Protocol: esp
        Mode: Tunnel
        Encryption Algorithm: aes256(key length=256 bits)
        Hash Algorithm: sha1
        Lifetime in seconds: 3600
        Lifetime in Kilobytes: 4608000
```

Step 10: Configure firewall policies to allow IKE negotiation through untrusted interface (applicable only if firewall license is also enabled):

```
Router1/configure# firewall internet
Router1/configure/firewall internet# policy 1000 in service ike self
Router1/configure/firewall internet/policy 1000 in# exit
Router1/configure/firewall internet# exit
```

Step 11: Display firewall policies in the internet map (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto
Action Advanced
---  --- -----------        ----------------   ----------------- ----
```

Step 12: Display firewall policies in the internet map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy internet detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Service Name is ike
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0
```

Step 13: Configure firewall policies for a group of mobile users to allow access to the local LAN (applicable only if firewall license is enabled):

```
Router1/configure# firewall corp
Router1/configure/firewall corp# policy 1000 in address 20.1.1.100
20.1.1.150 10.0.1.0 24
Router1/configure/firewall corp/policy 1000 in# exit
```

NOTE: The address range in this command typically matches the address range configured in the dynamic IKE policy (see Step 4).

Step 14: Display firewall policies in the corp map (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
1000 in  20.1.1.100         10.0.1.0/24        any   any   any   PERMIT E
         20.1.1.150
1022 out any                any                any   any   any   PERMIT SE
1023 in  any                any                any   any   any   PERMIT SE
1024 out any                any                any   any   any   PERMIT E
```

Step 15: Display firewall policies in the corp map in detail (applicable only if firewall license is enabled):

```
Router1# show firewall policy corp detail

Policy with Priority 1000 is enabled, Direction is inbound
Action permit, Traffic is transit
Logging is disable
Source Address is 20.1.1.100-20.1.1.150, Dest Address is 10.0.1.0/24
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1022 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1023 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0, Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is transit
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 11258, Bytes Out 5813
```

Step 16: Test the IPSec tunnel between the VPN client and the server by passing traffic from the client to the 10.0.1.0 network.

Step 17: After passing traffic through the tunnel, display the list of clients logged onto the VPN server and the IKE and IPSec SA tables:

```
Router1# show crypto dynamic clients
Client Address     Client Id               Policy
Advanced
--------------         ---------               ------
--------
192.168.107.105 david@abc-corp... sales:20.1.1.1        ModecfgGrp
```

```
Router1#show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------
sales      192.168.107.105 SA_MATURE  2052       pre-g1-3des-sha1
```

```
Router1# show crypto ike sa all detail

Crypto Policy name: sales
    Remote  ident david@abc-corp.com
    Peer Address is 192.168.107.105
    Transform: 3des, sha1, pre-shared-key
    DH Group: group1
    Bytes Processed 2052
    State is SA_MATURE
    Mode is Aggressive
    Life Time in Sec is unlimited, Life Time in Bytes is unlimited
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INsales    172.16.0.10xbba97427 840        esp-aes-sha1-tunl
sales      192.168.107.1050xcb0e23f3 560        esp-aes-sha1-tunl
```

```
Router1# show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
INsales    172.16.0.1  0xbba97427 840        esp-aes-sha1-tunl
sales      192.168.107.105 0xcb0e23f3 560        esp-aes-sha1-tunl
Router1#
Router1# show crypto ipsec sa all detail

Crypto Policy name: INsales
    Protocol is Any
    Local  ident(ip/mask/port): (20.1.1.1/255.255.255.255/any)
    Remote ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
    Peer Address is 172.16.0.1, PFS Group is disabled

inbound ESP sas
    Spi: 0xbba97427
    Transform: aes256 (key length=256 bits), sha1
    In use settings = {tunnel}
    Bytes Processed 840
    Hard lifetime in seconds 28750, Hard lifetime in kilobytes is
unlimited
    Soft lifetime in seconds 0, Soft lifetime in kilobytes is
unlimited

Crypto Policy name: sales
    Protocol is Any
    Local  ident(ip/mask/port): (10.0.1.0/255.255.255.0/any)
    Remote ident(ip/mask/port): (20.1.1.1/255.255.255.255/any)
    Peer Address is 192.168.107.105, PFS Group is disabled

outbound ESP sas
    Spi: 0xcb0e23f3
    Transform: aes256 (key length=256 bits), sha1
    In use settings = {tunnel}
    Bytes Processed 560
    Hard lifetime in seconds 28750, Hard lifetime in kilobytes is
unlimited
    Soft lifetime in seconds 28720, Soft lifetime in kilobytes is
unlimited
```

Configuring GRE

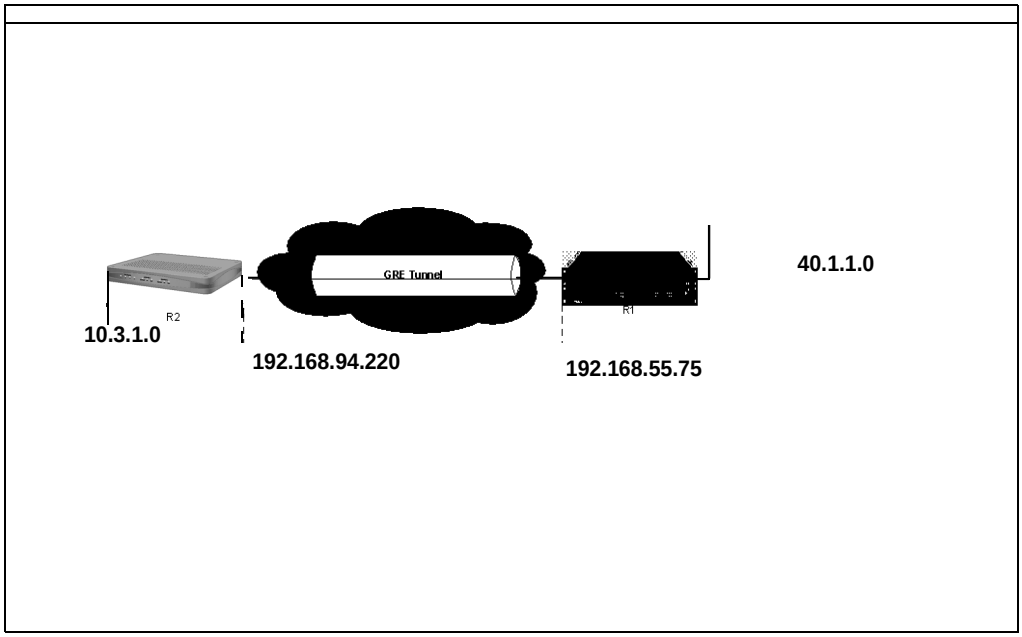
Generic Routing Encapsulation (GRE) is a standards-based (RFC1701, RFC2784) tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link between routers at remote points over an IP network. A tunnel is a logical interface that 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.

IPSec and GRE complement each other well, while IPSec provides a secure method of transporting data across the internet GRE provides the capability to transport routing protocols (for example: OSPF) that use broadcast and multicast.

GRE Configuration Examples

This example explains how to configure a basic GRE tunnel as shown in Figure 15.4.

Figure 15.4 Simple GRE configuration



Configuring Site to Site Tunnel

To configure GRE in a site to site tunnel configuration:

1. Configure the interface:

```
Foundry# configure terminal
Foundry/configure# interface bundle wan1
Foundry/configure/interface/bundle wan1# link t1 1
Foundry/configure/interface/bundle wan1# encapsulation ppp
Foundry/configure/interface/bundle wan1# ip address 192.168.94.220
255.255.255.0
Foundry/configure/interface/bundle wan1# exit
```

2. Configure the tunnel:

```
Foundry/configure# interface tunnel t0
Foundry/configure/interface/tunnel t0# ip 103.1.1.2 24
Foundry/configure/interface/tunnel t0# tunnel source 192.168.94.220
Foundry/configure/interface/tunnel t0# tunnel destination
192.168.55.75
Foundry/configure/interface/tunnel t0# exit
```

3. Configure the IP routes:

```
Foundry/configure# ip route 0.0.0.0 0.0.0.0 192.168.94.254
Foundry/configure# ip route 40.1.1.0 24 t0
```

NOTE: The peer of a local WAN interface cannot be used as a tunnel destination.

4. Verify that the tunnel is up and running. (If it is not, check the `Gateway` and `Source Address` fields.)

```
Foundry# show ip interface t0

t0 (unit number 5)
Type: TUNNEL
Flags: (0x74243) UP, RUNNING, MULTICAST-ROUTE
Internet Address: 103.1.1.2
Internet Netmask: 255.255.255.0
Internet Broadcast: 103.1.1.255
Maximum Transfer Unit: 1476 bytes
Source Address: 192.168.94.220
Destination Address: 192.168.55.75
Gateway: wan1
Protocol: GRE
Mac Address 00:50:52:60:00:00
```

For more information enter:

```
Foundry# show interface tunnel t0

Tunnel: t0 Status: up
Internet Address: 103.1.1.2 Internet Netmask: 255.255.255.0
Source Address: 192.168.94.220 Destination Address: 192.168.55.75
MTU: 1476 bytes                      Protocol: GRE
ICMP unreachable: will be sent       ICMP redirect: will be sent
Crypto Snet: not set               Protection: policy grecisco key
****
TTL: 30                              Keepalive: disabled
TOS: not set                         Path MTU discovery: disabled
Key Value: not set                   Checksum: disabled
Sequence Datagrams: disabled

Tunnel Statistics:
    Bytes Rx                   95112    Bytes Tx
60016
    Packets Rx                   860    Packets Tx
499
    Err Packets Rx                 0    Output Errs
0
```


5. Configure the Cisco side:

```
cisco > config t
cisco(config)#interface Ethernet2/0
cisco(config-if)#ip address 192.168.55.75255.255.255.0
cisco(config-if)#exit


cisco(config)#interface Tunnel 0
cisco(config-if)#ip address 103.1.1.1 255.255.255.0
cisco(config-if)#tunnel source 192.168.55.75
cisco(config-if)#tunnel destination 192.168.94.220
cisco(config-if)#exit

cisco(config)#ip route 0.0.0.0 0.0.0.0 192.168.55.254
cisco(config)#ip route 10.3.1.0 255.255.255.0 Tunnel0
```

With the tunnel properly configured and working, users on one side of the tunnel can ping users on the other side.

Configuring GRE Site to Site with IPSec

This example extends the first example by adding encryption to the tunnel.

1. Prepare the WAN link:

```
Foundry# configure terminal
Foundry/ configure# interface bundle wan1
Foundry/ configure/interface/bundle wan1# link t1 1
Foundry/ configure/interface/bundle wan1# encapsulation ppp
Foundry/ configure/interface/bundle wan1# ip address 192.168.94.220
255.255.255.0
Foundry/ configure/interface/bundle wan1# crypto untrusted
Foundry/ configure/interface/bundle wan1# exit
```

2. Configure the tunnel:

```
Foundry/ configure# interface tunnel t0
Foundry/ configure/interface/tunnel t0# ip address 103.1.1.2 24
Foundry/ configure/interface/tunnel t0# tunnel source
192.168.94.220
Foundry/ configure/interface/tunnel t0# tunnel destination
192.168.55.75
Foundry/ configure/interface/tunnel t0# tunnel protection grecisco
secretkeyfortest
Foundry/ configure/interface/tunnel t0# crypto untrusted
Foundry/ configure/interface/tunnel t0# exit
```

3. Configure the routes:

```
Foundry/ configure# ip route 0.0.0.0 0.0.0.0 192.168.94.254
Foundry/ configure# ip route 40.1.1.0 24 t0
```

4. Define the policy:

```
Foundry/ configure > firewall internet
Foundry/configure/firewall internet# policy 100 in proto gre self
Foundry/configure/firewall internet/policy 100 in# exit
Foundry/configure/firewall internet# policy 101 in service ike self
Foundry/configure/firewall internet/policy 101 in# exit 2
Foundry configure# firewall corp
Foundry/configure/firewall corp# policy 100 in self
```

5. Check the status of the tunnel by entering:

```
Foundry# show ip interface tunnel t0
```

Step 6:Validate the tunnel configuration by entering:

```
Foundry# show crypto ipsec policy all
```

Or enter:

```
Foundry# show crypto ike policy all
```

With the tunnel properly configured and working, users on one side of the tunnel can ping users on the other side.

Configuring GRE Site to Site with IPSec and OSPF

This example extends the previous IPSec configuration example by enabling Open Shortest Path First (OSPF) protocol which provides redundant paths for the tunnel.

1. To enable OSPF, add to the Foundry configuration above:

```
Foundry# configure terminal
Foundry/configure# router routerid 2.2.2.2
Foundry/configure# router ospf
Foundry/configure/router/ospf# interface t0 area 0
Foundry/configure/router/ospf# exit
```

2. Add to the Cisco configuration above:

```
cisco > config t
cisco(config)#router ospf 1
cisco(config-router)# network 103.1.1.0 0.0.0.255 area 0
```

3. To verify the OSPF configuration, enter:

```
Foundry# show ip ospf interface all
```


NOTE: Using the redistribute connected command adds a recursive route to the tunnel destination. This will cause the tunnel to shut down. To prevent this, add a 32-bit static route for the tunnel destination.

With the tunnel properly configured and working, users on one side of the tunnel can ping users on the other side.

Firewalls

Configuring firewalls allows administrators to adapt network protection policies to meet ever-changing hacker and intruder threats. Just as virus protection software requires updates to protect against the latest intrusion attacks, firewalls must be updated. In this release of Foundry software, administrators are able to filter traffic on specific ports, protect against Denial of Services attacks, enable IP packet reassembly, and so forth.

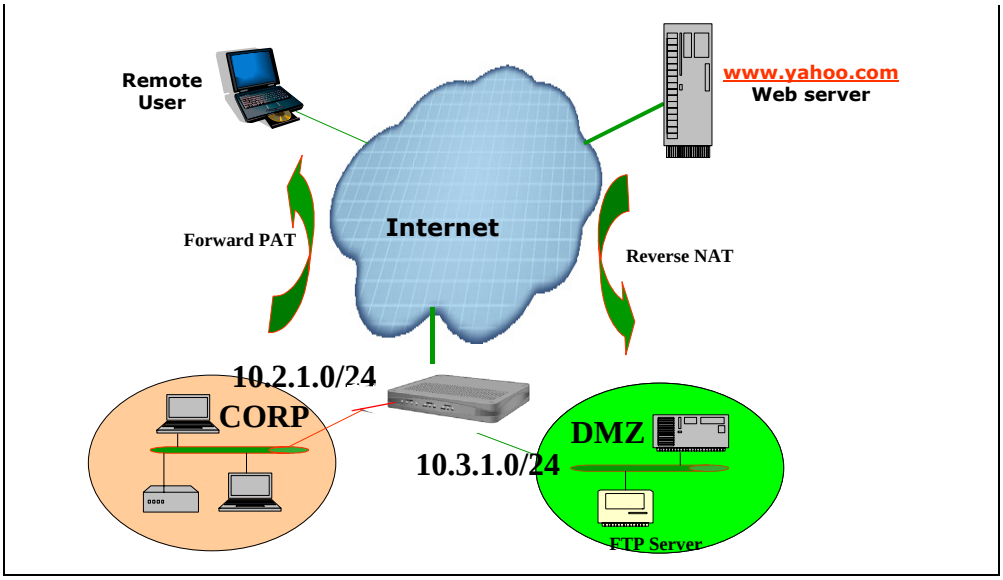
Because firewalls provide a barrier between the inside world—the corporate network, and the outside world—primarily the Internet, network administrators can further protect the network inside the firewall by using Network Address Translation (NAT). NAT allows users on the inside of the firewall to use private, nonroutable IP addresses which are translated to routable IP addresses at the firewall. The firewall manages the address translation by converting private IP addresses into a public address for outbound traffic. On inbound traffic the firewall converts traffic addressed to the public IP address into the various private IP addresses of the firewall-protected users. In addition to the protection of not being directly reachable from outside the network, the firewall-NAT enabled network conserves IP addresses.

Firewall Configuration Examples

Basic Firewall Configuration

Figure 15.5 illustrates the basic elements of a firewall. Refer to this illustration in the configuration example below.

Figure 15.5 Basic Firewall Configuration



A typical and basic firewall implementation is one which protects traffic to and from a network, a server farm, and the Internet. In this example, the firewall features in the Foundry router will protect the CORP network and the server farm in the DMZ from unauthorized access from the Internet.

To create this basic three-armed firewall configuration, complete these steps:

Step 1:Configure the Ethernet interfaces and the WAN interfaces with IP addresses:

```
Foundry/configure# interface ethernet 0
Configuring existing Ethernet interface
Foundry/configure/interface/ethernet 0# ip address 10.2.1.1 24
Foundry/configure/interface/ethernet 0# exit
Foundry/configure# interface ethernet 1
   Configuring existing Ethernet interface
Foundry/configure/interface/ethernet 1# ip address 10.3.1.1 24
Foundry/configure/interface/ethernet 1# exit
Foundry/configure# interface bundle wan
Foundry/configure/interface/bundle wan# link t1 1
Foundry/configure/interface/bundle wan# encapsulation p
Foundry/configure/interface/bundle wan# ip address 193.168.94.220 24
Foundry/configure/interface/bundle wan# exit
```

Step 2: Create the security zones CORP and DMZ and attach interfaces:

```
Foundry/configure# firewall corp
Foundry/configure/firewall corp# interface ethernet0
Foundry/configure/firewall corp# exit

Foundry/configure# firewall dmz
Foundry/configure/firewall dmz# interface ethernet1
Foundry/configure/firewall dmz# exit

Foundry/configure# firewall internet
Foundry/configure/firewall internet# interface wan
Foundry/configure/firewall internet# exit 2
```

Step 3: Verify that the interfaces are attached to the security zones:

```
Foundry/configure# show firewall interface all

Interface      Map Name
---------      --------
ethernet0      corp
ethernet1      dmz
wan            internet
```

Step 4: Create policies for Security Zone CORP that:

- Allow all outgoing traffic (with firewall policy priority 1024)
- Deny all incoming traffic (with firewall policy priority 1021)
- Create an object of type **http-filter** to block java traffic
- Modify policy 1024 to pat all outgoing traffic using public IP 193.168.94.220
- Modify policy 1024 to add a java HTTP filter.

```
Foundry/configure#
Foundry/configure/firewall corp#
Foundry/configure/firewall corp#
Foundry/configure/firewall corp# policy 1024 out
Foundry/configure/firewall corp/policy 1024 out# exit
Foundry/configure/firewall corp# policy 1021 in deny
Foundry/configure/firewall corp/policy 1021 in# exit
Foundry/configure/firewall corp# object
Foundry/configure/firewall corp/object# http-filter javadeny deny
*.java
Foundry/configure/firewall corp/object# exit
Foundry/configure/firewall corp# policy 1024 out nat-ip
193.168.94.220
Foundry/configure/firewall corp/policy 1024 out# apply-object http-
filter javadeny
Foundry/configure/firewall corp/policy 1024 out# exit
Foundry/configure/firewall corp# exit
```

Step 5: Verify the firewall policy for Security Zone CORP:

```
Foundry/configure# show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
1021 in  any                any                any   any   any   DENY   E
1022 out any                any                any   any   any   PERMIT SE
1023 in  any                any                any   any   any   PERMIT SE
1024 out any                any                any   any   any   PERMIT HNE
```

Step 6: Verify that the HTTP filter object in Security Zone CORP is created as configured:

```
Foundry/configure# show firewall object http-filter corp
Object Name     Action Log File Extensions
-----------     ------ --- ---------------
javadeny         deny   no  *.java
Foundry/configure#
```

Step 7: Create policies for Security Zone DMZ that:

- Create an object of type **nat-pool** with private IP address of FTP server
- Create an object of type **ftp-filter** to deny **put** and **mkdir** commands
- Create a firewall policy to allow inbound traffic to FTP server public IP address (193.168.94.221) of priority 100
- Modify policy 100 to add NAT pool object to translate incoming traffic for FTP server from public IP to private IP.
- Modify policy 100 to add an FTP filter.

```
Foundry/configure# firewall dmz
Foundry/configure/firewall dmz# object
Foundry/configure/firewall dmz/object# ftp-filter putdeny deny put
mkdir
Foundry/configure/firewall dmz/object# nat-pool ftpsrvr static
10.3.1.100
Foundry/configure/firewall dmz/object# exit
Foundry/configure/firewall dmz# policy 100 in address any any
193.168.94.221 32
Foundry/configure/firewall dmz/policy 100 in# apply-object nat-pool
ftpsrvr
Foundry/configure/firewall dmz/policy 100 in# apply-object ftp-filter
putdeny
Foundry/configure/firewall dmz/policy 100 in# exit
Foundry/configure/firewall dmz# exit
```

Step 8:Verify the firewall policy for Security Zone DMZ:

```
Foundry/configure# show firewall policy dmz
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action Advanced
---  --- -----------        ----------------   ----------------- ------ --------
100  in  any                193.168.94.221/32  any   any   any   PERMIT FNE
1022 out any                any                any   any   any   PERMIT SE
1023 in  any                any                any   any   any   PERMIT SE
1024 out any                any                any   any   any   PERMIT E
```

Step 9: Verify that the FTP filter objects for Security Zone DMZ are created as configured:

```
Foundry/configure# show firewall object ftp-filter dmz
Object Name     Action Log Commands
-----------     ------ --- --------
putdeny         deny   no  put mkdir
Foundry/configure#
```

Step 10: Create a default route out of the WAN:

```
Foundry/configure# ip route 0.0.0.0 0 wan
Foundry/configure#
```

Step 11:Verify the system configuration by displaying the running configuration:

```
Foundry/configure# show configuration running
Please wait... (up to a minute)


terminal
  exit terminal
qos
  exit qos
module  t1 1
    alarms
      thresholds
        exit thresholds
      exit alarms
    linemode
      exit linemode
    exit t1
module  t1 2
    alarms
      thresholds
        exit thresholds
      exit alarms
    linemode
      exit linemode
    exit t1
module  t1 3
    alarms
      thresholds
        exit thresholds
      exit alarms
    linemode
      exit linemode
    exit t1
module  t1 4
    alarms
      thresholds
        exit thresholds
      exit alarms
    linemode
      exit linemode
    exit t1
aaa
  tacacs
    retries 2
    time_out 5
    server_port 49
    exit tacacs
  radius
    exit radius
  exit aaa
interface  ethernet 0
    ip  address 10.2.1.1 255.255.255.0
    ip  multicast
        mode ospfrip2
        exit multicast
    mtu 4000
    icmp
      exit icmp
```



```
    qos
      exit qos
    vrrp_mode 0
    aaa
      exit aaa
    crypto trusted
    exit ethernet
interface  ethernet 1
    ip  address 10.3.1.1 255.255.255.0
    ip  multicast
        mode ospfrip2
        exit multicast
    mtu 4000
    icmp
      exit icmp
    qos
      exit qos
    vrrp_mode 0
    aaa
      exit aaa
    crypto trusted
    exit ethernet
interface  bundle wan
    link  t1 1
    encapsulation ppp
    ip  address 193.168.94.220 255.255.255.0
    ip  multicast ospfrip2
    red
      exit red
    icmp
      exit icmp
    qos
      exit qos
    aaa
      exit aaa
    crypto untrusted
    exit bundle
interface  console
    aaa
      exit aaa
    exit console
snmp
  system_id Foundry
  enable_trap
    exit enable_trap
  exit snmp
hostname Foundry
log utc
telnet_banner
  exit telnet_banner
event
  exit event
system  logging
    no console
    syslog
      host_ipaddr 193.168.94.35
      exit syslog
    exit logging
ip
```



```
  load_balance per_flow
  multicast
    exit multicast
  route 0.0.0.0 0.0.0.0 wan 1
  exit ip
policy  community_list
    exit community_list
crypto
  exit crypto
firewall global
  exit firewall
firewall internet
  interface wan
  policy 1024 out self
    exit policy
  exit firewall
firewall corp
  interface ethernet0
  object
    http-filter javadeny deny *.java
    exit object
  policy 1021 in deny
    exit policy
  policy 1022 out self
    exit policy
  policy 1023 in self
    exit policy
  policy 1024 out nat-ip 193.168.94.220
    apply-object http-filter javadeny
    exit policy
  exit firewall
firewall dmz
  interface ethernet1
  object
    nat-pool ftpsrvr static 10.3.1.100 10.3.1.100
    ftp-filter putdeny deny put mkdir
    exit object
  policy 100 in address any any 193.168.94.221 32
    apply-object ftp-filter putdeny
    apply-object nat-pool ftpsrvr
    exit policy
  policy 1022 out self
    exit policy
  policy 1023 in self
    exit policy
  policy 1024 out
    exit policy
  exit firewall
Foundry/configure#
```

Stopping DoS Attacks

The following commands show how to configure the firewall to defend against Denial of Service (DoS) attacks. Foundry provides protection against FTP bounce, ICMP error checks, IP sequence number checks, unaligned timestamps, MIME flooding, source routing checks, SYN flooding, and WIN nuke attacks. To configure the firewall for protection against all of these attacks, enter:

```
Foundry# config term
Foundry/configure# firewall global
Foundry/configure/firewall global# dos-protect
Foundry/configure/firewall global/dos-protect# enable-all
Foundry/configure/firewall global/dos-protect# exit 2
Foundry/configure#
```

Packet Reassembly

To configure the firewall to perform IP reassembly of oversized packets that have been fragmented, enter:

```
Foundry# config term
Foundry/configure# firewall global
Foundry/configure/firewall global# ip-reassembly
Foundry/configure/firewall global/ip-reassembly# fragment-count
100
Foundry/configure/firewall global/ip-reassembly# fragment-size 56
Foundry/configure/firewall global/ip-reassembly# packet-size 2048
Foundry/configure/firewall global/ip-reassembly# timeout 20
Foundry/configure/firewall global/ip-reassembly# exit 2
Foundry/configure#
```

NAT Configurations

Network Address Translation (NAT) was defined to serve two purposes:

- Allowed LAN administrators to create secure, private, non-routable IP networks behind firewalls
- Stretched the number of available IP addresses by allowing LANs to use one public (real) IP address as the gateway with a very large pool of NAT addresses behind it.

In the most common NAT application (which is to provide secure networking behind a firewall), the device (Foundry system) that connects the user LAN to the Internet will have two IP addresses:

- A private IP address on the LAN side for the RFC 1918 address range
- A public address, routable over the Internet, on the WAN side

Consider a PC on the LAN sending a packet destined for *some.server.com*. The source IP address and port are in the packet together with the destination IP address and port. When the packet arrives at the Foundry system it will be de-encapsulated, modified, and re-encapsulated. The re-encapsulated packet sent by the Foundry system destined for the Internet contains the Foundry system's public IP address, a source port allocated from its list of available ports, and the same destination IP address and port number generated by the PC. The Foundry system also adds an entry into a table it keeps, which maps the internal address and source port number that the PC generated against the port number it allocated to this session. Therefore, when *some.server.com* sends a reply packet to the PC, the Foundry system can quickly determine how it needs to re-write the packet before transmitting it back on to the LAN.

Dynamic NAT is used when packets destined for the Internet are transported from a LAN using the public source IP address assigned to the local router. Dynamic NAT performs this task well, but it does not permit providing services to the Internet from inside a LAN which requires the use of static NAT. Static NAT also requires a public address from the upstream service provider. Individual PCs within a LAN are assigned RFC 1918 reserved IP addresses to enable access to other PCs within the LAN. The Foundry system is configured with static mapping, which maps the internal RFC 1918 IP addresses for each PC to the appropriate public IP address. When traffic is sent to the public address listed in the static mapping, the Foundry system forwards the packets to the correct PC within the LAN, according to the mapping relationship established.

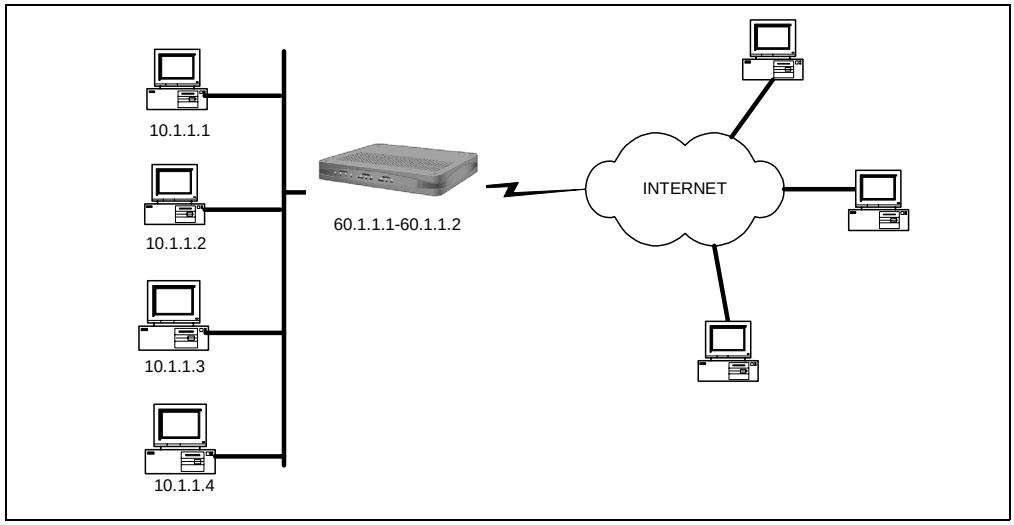
NAT Configuration Examples

Dynamic NAT (many to many)

In dynamic (many-to-many) NAT type, multiple source IP addresses in the corporate network will be mapped to multiple NAT IP addresses (not necessarily of equal number). For a set of local IP address from 10.1.1.1 to 10.1.1.4 there will be a set of NAT IP address from 60.1.1.1 to 60.1.1.2. In case of many-to-many NAT, only IP address translation takes place, i.e., if a packet travels from 10.1.1.1 to yahoo.com, Foundry-Firewall only substitutes the source address in the IP header with one of the NAT IP address and the source port will be the same as the original. If traffic emanates from the same client to any other server, the same NAT IP address is assigned. The advantage is that the NAT IP addresses are utilized in a better and optimum manner dynamically.

If a NAT IP address cannot be allocated dynamically at the connection creation time, the packet would be dropped.

Figure 15.6 Dynamic NAT



The dynamic NAT configuration shown in includes:

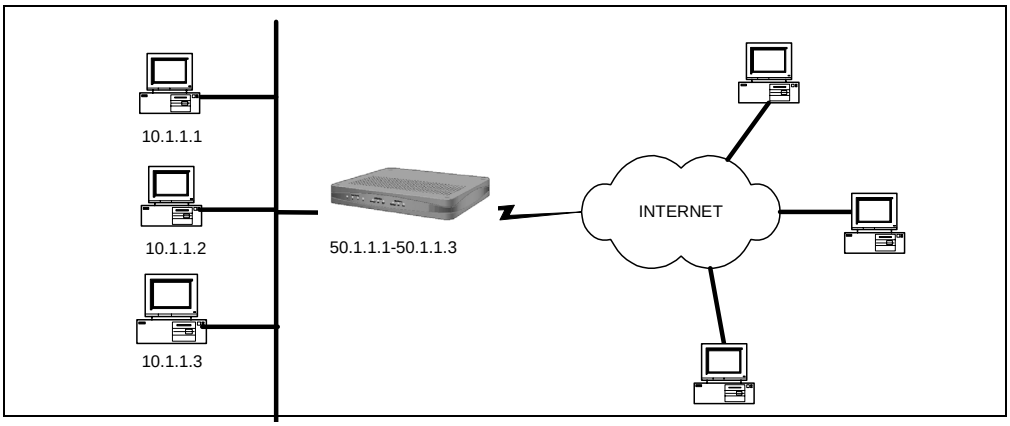
- Private network addresses:10.1.1.1—10.1.1.4
- Public (NAT) IP address range: 60.1.1.1—60.1.1.2

To create NAT pool with type **dynamic,** specify the IP address and the NAT ending IP address.Then add a policy with the source IP address range, and attach the NAT pool to the policy.

```
Foundry/configure# firewall corp
Foundry/configure/firewall corp# object
Foundry/configure/firewall corp/object# nat-pool addresspoolDyna
dynamic 60.1.1.1 60.1.1.2
Foundry/configure/firewall corp/object# exit
Foundry/configure/firewall corp# policy 8 out address 10.1.1.1
10.1.1.4 any any
Foundry/configure/firewall corp/policy 8 out# apply-object nat-
pool addresspoolDyna
Foundry/configure/firewall corp/policy 8 out# exit 2
Foundry/configure#
```

Static NAT (one to one)

Figure 15.7 Static NAT



In static (one-to-one) NAT type, for each IP address in the corporate network, one NAT IP address will be used. For example, for the three IP addresses from 10.1.1.1 to 10.1.1.3, there is a set of three NAT IP address from 50.1.1.1 to 50.1.1.3. In case of one-to-one NAT, only IP address translation takes place, that is, if a packet travels from 10.1.1.1 to yahoo.com, the Foundry-Firewall only substitutes the source address in the IP header with the NAT IP address. The source port will be the same as the original.

The static NAT configuration shown in Figure 15.7 includes:

- Private network address:10.1.1.1—10.1.1.3
- Public (NAT) IP address range: 50.1.1.1—50.1.1.3

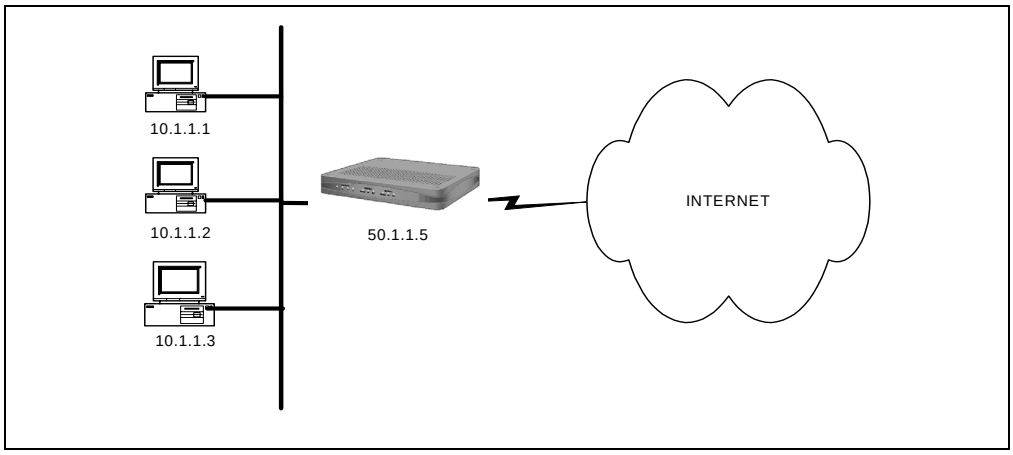
To create NAT pool with type **static**, specify the IP address and the ending NAT IP address. Add a policy with source IP address range and attach NAT pool to the policy.

```
Foundry/configure# firewall corp
Foundry/configure/firewall corp object
Foundry/configure/firewall corp/object# nat-pool addresspoolStat
static 50.1.1.1 50.1.1.3
Foundry/configure/firewall corp/object# exit
Foundry/configure/firewall corp# policy 7 out address 10.1.1.1
10.1.1.3 any any
Foundry/configure/firewall corp/policy 7 out# apply-object nat-
pool addresspoolStat
Foundry/configure/firewall corp/policy 7 out# exit 2
Foundry/configure#
```

Port Address Translation (Many to one)

NAT allows multiple IP addresses to be mapped to one address.

Figure 15.8 Mapping Multiple NAT Addresses to One Public IP Address



There are two methods to configure Port Address Translation (PAT) on the Foundry gateway. In the first method, specify the IP address to the `nat-ip` parameter in the **`policy`** command. In the second method, create a pool of type PAT and then attach it to the policy.

In PAT, multiple hosts can share the same IP address.

The PAT configuration shown in Figure 15.8 includes:

- Private network address: 10.1.1.1—10.1.1.3
- PAT address: 50.1.1.5

Method:1 – Specifying NAT address with the policy command

To configure this method of PAT, add the policy with the source IP address range, then specify the `nat-ip` address in the **`policy`** command:

```
Foundry/configure# firewall corp
Foundry/configure/firewall corp# policy 2 out address 10.1.1.1
10.1.1.3 any any nat-ip 50.1.1.5
Foundry/configure/firewall corp/policy 2 out# exit 2
Foundry/configure#
```

Method:2 – Attaching nat pool to the policy

To configure the second type of NAT, create a NAT pool with type **`pat`** and specify the IP address. Then add the policy with the source IP address range. Finally, attach the NAT pool to the policy.


```
Foundry/configure# firewall corp
Foundry/configure/firewall corp# object
Foundry/configure/firewall corp/object# nat-pool addresspoolPat
pat 50.1.1.5
Foundry/configure/firewall corp/object# exit
Foundry/configure/firewall corp# policy 2 out address 10.1.1.1
10.1.1.3 any any
Foundry/configure/firewall corp/policy 2 out# apply-object nat-
pool addresspoolPat
Foundry/configure/firewall corp/policy 2 out# exit 2
   Foundry/configure#
```

Security Protocol Defaults

This section provides information about IPSec supported protocols and modes, encryption algorithms and block sizes, and Foundry IPSec and IKE default values.

IPSec Supported Protocols and Algorithms

The following tables provide supported protocol and algorithm information.

Table 15.1: IPSec Protocols Support

Supported Security Protocols	Mode
ESP	Tunnel Transport
AH	Tunnel Transport

Table 15.2: Encryption Algorithms

Encryption Algorithms for ESP	**Block Size**
Data Encryption Standard (DES)	56 bits
Triple Data Encryption Standard (3DES)	168 bits
Advanced Encryption Standard (AES-128)	128 bits
Advanced Encryption Standard (AES-192)	192 bits
Advanced Encryption Standard (AES-256)	256 bits
Null Encryption	

Table 15.3: Authentication Algorithms

Authentication Algorithms for AH/ ESP	Hash Size
HMAC-MD5-96	96 bits
HMAC-HSHA1-96	96 bits

Table 15.4: Diffie-Hellman Groups

Diffie-Hellman Groups for Authentication	Key Size
Group 1	768 bits
Group 2	1024 bits
Group 5	1536 bits

Foundry IKE and IPSec Defaults

To minimize configuration required by the user, default IKE and IPSec values have been implemented in Foundry's encryption scheme. Foundry supports a maximum of 100 IPSec tunnels.

IKE Defaults

Table 15.5: lists IKE defaults. When the user creates an IKE policy specifying an IKE peer, an IKE proposal with priority 1 is automatically created. However, to make the IKE policy fully functional, the user must enter a pre-shared key.

Table 15.5: IKE Default Values

Parameter Name	Foundry Default Value: Site to Site	Foundry Default Value: Remote Access
Mode	Main mode	Aggressive mode
Perfect forward secrecy	Disabled	Disabled
Hash algorithm	SHA1	SHA1
Encryption algorithm	DES	DES
Authentication method	PreShared	PreShared
DH Group	Group 1	Group 1
Lifetime	86400 seconds	86400 seconds
Response type	Initiator and responder	Responder only

IPSec Defaults

Table 15.6: lists IPSec defaults. When the user creates an IPSec policy and provides the match address, an IPSec proposal with priority 1 is automatically created. When an outbound policy is specified, an inbound policy is automatically created.

Table 15.6: IPSec Default Values

Parameter Name	Foundry Default Value: Site to Site and Remote Access
Key management type	Automatic
Hash algorithm	SAH1
Encryption algorithm	3DES
Protocol	ESP
Mode	Tunnel
Lifetime in seconds	3600 seconds
Lifetime in kilobytes	4608000
Direction	Out
Position in SPD where policy added	End
Perfect forward secrecy	Disabled

Firewall Default Values

This section provides information about firewall default values. Each security zone can have a maximum of 1024 policies ranging from 1—1024. The maximum number of security zones supported is 25.

Table 15.7: Firewall Default Policies by Security Zone

Security Zone	Incoming Firewall Policy for Transit Traffic	Outgoing Firewall Policy for Transit Traffic	Incoming Firewall Policy for Self Traffic	Outgoing Firewall Policy for Self Traffic
Corp	Deny All (Implicit)	Permit All (Priority 1024)	Permit All (Priority 1022)	Permit All (Priority 1023)
User Created Security Zone	Deny All	Permit All (Priority 1024)	Permit All (Priority 1022)	Permit All (Priority 1023)
Internet	N/A	N/A	Deny All	Permit All (Priority 1024)

Table 15.8: Firewall per policy defaults

Policy Parameter	Default Value
Priority	No Default

Table 15.8: Firewall per policy defaults

Direction	No Default
Action	Permit
Traffic type	Transit
Source Port	Any
Destination Port	Any
Schedule	Disabled
FTP Filter	Disabled
SMTP Filter	Disabled
HTTP Filter	Disabled
RPC Filter	Disabled
NAT	Disabled
Maximum Connections	1024
Connection Rate	Disabled
Policing	Disabled
Bandwidth	Disabled

Table 15.9: Default Connection Limit by Security Zone

Security Zone	**Maximum Connections Default**
Corp	1024 outgoing connections
User Created Security Zone	1024 outgoing connections
Internet	3072
Self	216
Internet to Self	108

Table 15.10: DoS Protection Defaults (Configured DoS Attacks)

Security Zone	**Maximum Connections Default**
Syn Flooding Attack Check	Enabled
ICMP Error Attack Check	Enabled
Source Route Attack Check	Disabled

Table 15.10: DoS Protection Defaults (Configured DoS Attacks)

Win Nuke Attack Check	Disabled
IP Unaligned Time stamp check	Disabled
TCP Sequence Number Prediction Check	Disabled
TCP Sequence Number Range Check	Disabled
FTP Bounce Check	Disabled

Tunneling Default Values

This section provides the IP-IP and GRE tunneling protocol default values.

Table 15.11: Tunnel Interface Defaults

Parameter	Default Value
IP Address	No Default
Tunnel Source	No Default
Tunnel Destination	No Default
MTU	1476 - Not configurable
ICMP unreachable	Enabled
ICMP redirect	Enabled
Crypto/Tunnel Protection	Disabled
Tunnel TTL	30
Keepalive	Disabled
Tunnel Mode	GRE
Tunnel TOS	Copy from Inner
Tunnel Path Mtu Discovery	Disabled
Tunnel Sequence	Disabled
Tunnel Checksum	Disabled
Tunnel Key	Disabled
Shutdown	Disabled

Index