DIGITAL GIGAswitch/Router

User Reference Manual

Part Number: 9032684

June 1998

This manual describes how to configure and monitor the DIGITAL GIGAswitch/Router.

Revision/Update Information: This is a new document.

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.

Cabletron Systems, Inc. Program License Agreement

IMPORTANT: Before utilizing this product, carefully read this License Agreement.

This document is an agreement between you, the end user, and Cabletron Systems, Inc. ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software (the "Program") contained in this package. The Program may be contained in firmware, chips, or other media. BY UTILIZING THE ENCLOSED PRODUCT, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE LICENSE AND THE LIMITATION OF WARRANTY AND DISCLAIMER OF LIABILITY. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

Cabletron Software Program License

1. LICENSE. You have the right to use only the one (1) copy of the Program provided in this package subject to the terms and conditions of this License Agreement.

   You may not copy, reproduce, or transmit any part of the Program except as permitted by the Copyright Act of the United States or as authorized by Cabletron.

2. OTHER RESTRICTIONS. You may not reverse engineer, decompile, or disassemble the Program.

3. APPLICABLE LAW. This License Agreement shall be interpreted and governed under the laws and in the state and federal courts of New Hampshire. You accept the personal jurisdiction and venue of the New Hampshire courts.

Exclusion of Warranty and Disclaimer of Liability

1. EXCLUSION OF WARRANTY. Except as may be specifically provided by Cabletron in writing, Cabletron makes no warranty, expressed or implied, concerning the Program (including its documentation and media).

   CABLETRON DISCLAIMS ALL WARRANTIES, OTHER THAN THOSE SUPPLIED TO YOU BY CABLETRON IN WRITING, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PROGRAM, THE ACCOMPANYING WRITTEN MATERIALS, AND ANY ACCOMPANYING HARDWARE.

2. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL CABLETRON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR RELIANCE DAMAGES, OR OTHER LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THIS CABLETRON PRODUCT, EVEN IF CABLETRON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR ON THE DURATION OR LIMITATION OF IMPLIED WARRANTIES, IN SOME INSTANCES THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

United States Government Restricted Rights

The enclosed product (a) was developed solely at private expense; (b) contains “restricted computer software” submitted with restricted rights in accordance with Section 52227-19 (a) through (d) of the Commercial Computer Software - Restricted Rights Clause and its successors, and (c) in all respects is proprietary data belonging to Cabletron and/or its suppliers.

For Department of Defense units, the product is licensed with “Restricted Rights” as defined in the DoD Supplement to the Federal Acquisition Regulations, Section 52.227-7013 (c) (1) (ii) and its successors, and use, duplication, disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Cabletron Systems, Inc., 35 Industrial Way, Rochester, New Hampshire 03867-0505.

Contents

Preface About This Manual

Chapter 1 DIGITAL GIGAswitch/Router Product Overview

Chapter 2 Bridging Configuration Guide

Chapter 3 IP Routing Configuration Guide

Chapter 4 RIP Configuration Guide

Chapter 5 OSPF Configuration Guide

Chapter 6 Routing Policy Configuration Guide

Chapter 7 Multicast Routing Configuration Guide

Chapter 10 QoS Configuration Guide

Chapter 11 Performance Monitoring Guide

About This Manual

Purpose of This Manual

This manual provides detailed information and procedures for configuring the 8-slot DIGITAL GIGAswitch/Router (DGSRA-AA) software. If you have not yet installed the DIGITAL GIGAswitch/Router software, use the instructions in the *DIGITAL GIGAswitch/Router Getting Started Guide* to install the chassis and perform basic setup tasks, then return to this manual for more detailed configuration information.

Intended Audience

Read this manual if you are a network administrator responsible for configuring and monitoring the GIGAswitch/Router (GSR).

Organization

This manual is organized as follows:

If You Want To...	See...
Read overview information	Chapter 1
Configure bridging	Chapter 2
Configure IP interfaces and global routing parameters	Chapter 3
Configure RIP routing	Chapter 4
Configure OSPF routing	Chapter 5
Configure Routing Policies	Chapter 6
Configure IP Multicast routing	Chapter 7
Configure IPX routing	Chapter 8
Configure filters	Chapter 9
Configure QoS (Quality of Service) parameters	Chapter 10
Monitor performance	Chapter 11

Associated Documentation

The DIGITAL documentation set includes the following items. Refer to these other documents to learn more about your product.

For Information About...	See the...
Installing and setting up the GIGAswitch/Router	*DIGITAL GIGAswitch/Router Getting Started Guide*
Managing the GIGAswitch/Router using the DIGITAL clearVISN element management application	*DIGITAL clearVISN CoreWatch User's Guide* and the DIGITAL clearVISN CoreWatch online help
The complete syntax for all CLI commands	*DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*
System messages and SNMP traps	*DIGITAL GIGAswitch/Router Error Message Reference Manual*

Correspondence

Documentation Comments

If you have comments or suggestions about this document, send them to the Network Products Engineering.

Attn.:Documentation Project Manager
E-MAIL:doc_quality@lkg.mts.dec.com

Online Services

Further product information is available on the DIGITAL Network Products Business World Wide Web Sites. All sites maintain the same, rich set of up-to-date information on products, technologies and programs.

The Web Sites can be reached at the following geographic locations:

Americas:	http://www.networks.digital.com
Europe:	http://www.networks.europe.digital.com
Asia Pacific:	http://www.networks.digital.com.au

Chapter 1 DIGITAL GIGAswitch/Router Product Overview

The 8-slot DIGITAL GIGAswitch/Router (DGSRA-AA) provides non-blocking, wire-speed Layer-2 (switching), Layer-3 (routing) and Layer-4 (application) switching. The hardware provides wire-speed performance regardless of the performance monitoring, filtering, and Quality of Service (QoS) features enabled by the software. You do not need to accept performance compromises to run QoS or access control lists (ACLs).

The following table lists the basic hardware and software specifications for the 8-slot GIGAswitch/Router (DGSRA-AA).

Feature	Specification
Throughput	• 16-Gbps non-blocking switching fabric • 15 million packets-per-second routing throughput
Capacity	• Up to 250,000 routes • Up to 2,000,000 Layer-4 application flows • 400,000 Layer-2 MAC addresses • 4,096 Virtual LANs (VLANs) • 20,000 Layer-2 security and access-control filters • 3MB input/output buffering per Gigabit port • 1MB input/output buffering per 10/100 port
Routing protocols	• IP: RIPv1/v2, OSPF • IPX: RIP, SAP • Multicast: IGMP, DVMRP
Bridging and VLAN protocols	• 802.1d Spanning Tree • 802.1Q (VLAN trunking)
Media Interface protocols	• 802.3 (10Base-T) • 802.3u (100Base-TX, 100BASE-FX) • 802.3x (1000Base-SX, 1000Base-LX) • 802.3z (1000Base-SX, 1000Base-LX)

Feature	Specification
Quality of Service (QoS)	• Layer-2 prioritization (802.1p) • Layer-3 source-destination flows • Layer-4 source-destination flows • Layer-4 application flows
RMON	• RMONv1/v2 for each port
Management	• SNMP • CoreWatch Element Manager (GUI) • Emacs-like Command Line Interface (CLI)
Port mirroring	• Traffic to Control Module • Traffic from specific ports • Traffic to specific chassis slots (line cards)
Hot swapping	• Power supply (when redundant supply is installed and online)
Redundancy	• Redundant and hot-swappable power supplies

Supported Media (Encapsulation Type)

The GIGAswitch/Router supports the following industry-standard networking media:

- IP: IEEE 802.3 SNAP and Ethernet Type II
- IPX: IEEE 802.3 SNAP, Ethernet Type II, IPX 802.3, 802.2
- 802.1Q VLAN Encapsulation

Supported Routing Protocols

The GIGAswitch/Router supports many routing protocols based on open standards. The GIGAswitch/Router can receive and forward packets concurrently from any combination of the following:

- Interior Gateway Protocols
  - Open Shortest Path First (OSPF) Version 2

  - Routing Information Protocol (RIP) Version 1, 2

“IP Routing Configuration Guide” on page 3 - 1 describes these protocols in detail.

The GIGAswitch/Router supports the following Novell IPX routing protocols:

- Routing Information Protocol (RIP)
- Service Advertising Protocol (SAP)

“IPX Routing Configuration Guide” on page 8 - 1 describes these protocols in detail.

Configuring the DIGITAL GIGAswitch/Router

The GIGAswitch/Router provides a command line interface (CLI) that allows you to configure and manage the GIGAswitch/Router. The CLI has several command modes, each of which provides a group of related commands that you can use to configure the GIGAswitch/Router and display its status. Some commands are available to all users; others can be executed only after the user enters an “Enable” password.

You use the CLI to configure ports, IP/IPX interfaces, routing, switching, security filters and Quality of Service (QoS) policies.

Understanding the Command Line Interface

The GIGAswitch/Router Command Line Interface (CLI) provides access to several different command modes. Each command mode provides a group of related commands. This chapter describes how to access and list the commands available in each command mode and explains the primary uses for each command mode. This chapter also describes the other features of the user interface.

GIGAswitch/Router commands can be entered at a terminal connected to the access server or router using the command line interface (CLI). The GIGAswitch/Router can also be configured using the CoreWatch Java-based management application. Using CoreWatch is described in the *DIGITAL clearVISN CoreWatch User’s Guide*.

Basic Line Editing Commands

The CLI supports EMACs-like line editing commands. The following table lists some commonly used commands.

Key sequence	Command
Ctrl-A	Move cursor to beginning of line
Ctrl-B	Move cursor back one character

Key sequence	Command
Ctrl-D	Delete character
Ctrl-E	Move cursor to end of line
Ctrl-F	Move cursor forward one character
Ctrl-N	Scroll to next command in command history (use the **`cli show history`** command to display the history)
Ctrl-P	Scroll to previous command in command history
Ctrl-U	Erase entire line
Ctrl-X	Erase from cursor to end of line
Ctrl-Z	Exit current access mode to previous access mode

Access Modes

The GIGAswitch/Router CLI has four access modes.

- **User** – Allows you to display basic information and use basic utilities such as ping but does not allow you to display SNMP, filter and access control list information or make other configuration changes. You are in User mode when the command prompt ends with this character:
  `>`
- **Enable** – Allows you to display SNMP, filter, and access control information as well as all the information you can display in User mode. To enter Enable mode, enter the **`enable`** command, then supply the password when prompted. When you are in Enable mode, the command prompt ends with this character:
  #
- **Configure** – Allows you to make configuration changes. To enter Configure mode, first enter Enable mode (**`enable`** command), then enter the **`configure`** command from the Enable command prompt. When you are in Configure mode, the command prompt ends with these characters:
  `(config)#`
- **Boot** – This mode appears when the GIGAswitch/Router the external flash card or the system image is not found during bootup. You should enter the **reboot** command

to reset the GIGAswitch/Router. If the GIGAswitch/Router still fails to bootup, please contact your DIGITAL representative.

Note: The command prompt will show the name of the DIGITAL GIGAswitch/ Router in front of the mode character(s). The default name is “gs/r”.

When you are in Configure or Enable mode, use the **`exit`** command or press Ctrl-z to exit to the previous access mode.

Note: When you exit Configure mode, the CLI will ask you whether you want to activate the configuration commands you have issued. If you enter **`Y`** (Yes), the configuration commands you issued are placed into effect and the DIGITAL GIGAswitch/Router’s configuration is changed accordingly. However, the changes are not written to the Startup configuration file in the Control Module’s boot flash and therefore are not reinstated after a reboot.

User Mode

After you log in to the GIGAswitch/Router, you are automatically in User mode. The User commands available are a subset of those available in Enable mode. In general, the User commands allow you to display basic information and use basic utilities such as ping information.

To list the User commands, enter:

List the User commands.	**?**

The User mode command prompt consists of the GIGAswitch/Router name followed by the angle bracket (>):

```
gs/r>
```

The default name is GS/R unless it has been changed during initial configuration using the system set name command. Refer to the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual* for information on the system facility.

To list the commands available in User mode, enter a question mark (?) as shown in the following example:

```
gs/r> ?
 aging               - Show L2 and L3 Aging information
 cli                 - Modify the command line interface behavior
 dvmrp               - Show DVMRP related parameters
 enable              - Enable privileged user mode
 exit                - Exit current mode
```

```
file               - File manipulation commands
igmp               - Show IGMP related parameters
ipx                - Show IPX related parameters
l2-tables          - Show L2 Tables information
logout             - Log off the system
multicast          - Configure Multicast related parameters
ping               - Ping utility
statistics         - Show or clear GSR statistics
stp                - Show STP status
traceroute         - Traceroute utility
vlan               - Show VLAN-related parameters
```

Enable Mode

Enable mode provides more facilities than User mode. You can display critical features within Enable mode including router configuration, access control lists and SNMP statistics. To enter Enable mode, enter the **`enable`** command, then supply the password when prompted.

To list the Enable commands, enter:

List the user Enable commands.	**`?`**

The Enable mode command prompt consists of the GIGAswitch/Router name GS/R followed by the pound sign (#):

```
gs/r#
```

To list the commands available in Enable mode, enter a question mark (?) as shown in the following example:

```
gs/r# ?
    acl               - Show L3 Access Control List
    aging             - Show L2 and L3 Aging information
    arp               - Show or modify ARP entries
    cli               - Modify the command line interface
                        behavior
    configure         - Enter Configuration Mode
    copy              - Copy configuration database
    dvmrp             - Show DVMRP related parameters
    enable            - Enable privileged user mode
    exit              - Exit current mode
    file              - File manipulation commands
    filters           - Show L2 security filters
    http              - Show http parameters
    igmp              - Show IGMP related parameters
    interface         - Show interface related parameters
```

```
ip                  - Show IP related parameters
ip-router           - Show unicast IP Routing related
                      parameters
ipx                 - Show IPX related parameters
l2-tables           - Show L2 Tables information
logout              - Log off the system
mtrace              - Multicast Traceroute utility
multicast           - Configure Multicast related parameters
ospf                - Show/Monitor Open Shortest Path First
                      Protocol (OSPF).
ping                - Ping utility
port                - Show or change Port parameters
qos                 - Show Quality of Service parameters
reboot              - Reboot the system
rip                 - Show/Query Routing Information Protocol
                      (RIP) tables
snmp                - Show SNMP related parameters.
statistics          - Show or clear GSR statistics
stp                 - Show STP status
system              - Show system-wide parameters
tacacs              - Show TACACS related parameters
traceroute          - Traceroute utility
vlan                - Show VLAN-related parameters
```

To exit Enable mode and return to User mode, use one of the following commands:

Exit Enable mode.	**exit** **Ctrl-Z**

Configure Mode

Configure mode provides the capabilities to configure all features and functions on the GIGAswitch/Router. You can configure features and functions within Configure mode including router configuration, access control lists and spanning tree.

To list the Configure commands, enter:

List the Configure commands.	**?**

The Configure mode command prompt consists of the GS/R name followed by the pound sign (#):

```
gs/r(config)#
```

To list the commands available in Configure mode, enter a question mark (**?**) as shown in the following example:

```
gs/r(config)# ?
 acl                  - Configure L3 Access Control List
 acl-edit             - Edit an ACL in the ACL Editor
 aging                - Configure L2 and L3 Aging
 arp                  - Configure ARP entries
 bgp                  - Configure Border Gateway Protocol (BGP)
 cli                  - Modify the command line interface behavior
 dvmrp                - Configure DVMRP related parameters
 exit                 - Exit current mode
 filters              - Configure L2 security filters
 http                 - Configure SNMP related parameters.
 igmp                 - Configure IGMP related parameters
 interface            - Configure interface related parameters
 ip                   - Configure IP related parameters
 ip-router            - Configure Unicast Routing Protocol related
                         parameters
 ipx                  - Configure IPX related parameters
 ospf                 - Configure Open Shortest Path Protocol (OSPF)
 port                 - Configure Port parameters
 qos                  - Configure Quality of Service parameters
 rip                  - Configure Routing Information Protocol (RIP)
 snmp                 - Configure SNMP related parameters.
 stp                  - Configure STP parameters
 system               - Configure system-wide parameters
 tacacs               - Configure TACACS related parameters
 vlan                 - Configure VLAN-related parameters

Special configuration mode commands:
 erase                - Erase configuration information
 negate               - Negate a command or a group of commands
                       using line numbers
 no                   - Negate matching commands
 save                 - Save configuration information
 search               - Look up a command in configuration
 show                 - Show configuration commands
```

To exit Configure mode and return to Enable mode, use one of the following commands:

Exit Configure mode.	**exit** **Ctrl-Z**

Boot PROM Mode

If your GIGAswitch/Router does not find a valid system image on the external PCMCIA flash, the system might enter programmable read-only memory (PROM) mode. You should then reboot the GIGAswitch/Router at the boot PROM to restart the system. If the system fails to reboot successfully, please contact your DIGITAL representative to resolve the problem.

To reboot the GIGAswitch/Router from the ROM monitor mode, enter the following command.

Reboot in Boot PROM mode.	**`reboot`**

Disabling a Function or Feature

The CLI provides for an implicit negate. This allows for the “disabling” of a feature or function which has been “enabled”. Use the **`negate`** command on a specific line of the active configuration to “disable” a feature or function which has been enabled. For example, Spanning Tree Protocol is disabled by default. If after enabling Spanning Tree Protocol on the DIGITAL GIGAswitch/Router, you want to disable STP, you must specify the **`negate`** command on the line of the active configuration containing the **`stp enable`** command.

Loading System Images and Configuration Files

The GIGAswitch/Router contains an internal flash on the Control Module and an external PC flash. The internal flash contains the GIGAswitch/Router boot image and user defined configuration files. An external PC flash contains the system image executed by the Control module. When an GIGAswitch/Router boots, the boot image is executed first, followed by the system image and finishing with a configuration file.

Boot and System Image

Only one boot image exists on the internal flash of the GIGAswitch/Router Control Module. Multiple system images can be stored on the external PC flash.

Configuration Files

The GIGAswitch/Router uses three special configuration files:

- **Active** – The commands from the Startup configuration file and any configuration commands that you have made active from the scratchpad (see below).

Caution: The active configuration remains in effect only during the current power cycle. If you power down or reboot the GIGAswitch/Router without saving the active configuration changes to the Startup configuration file, the changes are lost.

- **Startup** – The configuration file that the GIGAswitch/Router uses to configure itself when the system is powered on.
- **Scratchpad** – The configuration commands you have entered during a

management session. These commands do not become active until you explicitly activate them. Because some commands depend on other commands for successful execution, the GIGAswitch/Router scratchpad simplifies system configuration by allowing you to enter configuration commands in any order, even when dependencies exist. When you activate the commands in the scratchpad, the GIGAswitch/Router sorts out the dependencies and executes the command in the proper sequence.

Loading System Image Software

By default, the GIGAswitch/Router boots using the system image software installed on the Control Module's PCMCIA flash card. To upgrade the system software and boot using the upgraded image, use the following procedure.

1. Display the current boot settings by entering the following command:

   system show version

   Here is an example:

```
gs/r# system show version
Software Information
  Software Version   : 1.0.1
  Copyright         : Copyright (c) 1996-1998 Cabletron Systems, Inc.
  Image Information  : Version 1.0.1, built on Fri Jun 12 19:28:49 1998
       Image Boot Location: file:/pc-flash/boot/gsr1010/
```

Note: In this example, the location "pc-flash" indicates that the GIGAswitch/ Router is set to use the factory-installed software on the flash card.

2. Copy the software upgrade you want to install onto a TFTP server that the GIGAswitch/Router can access. (Use the **ping** command to verify that the GIGAswitch/Router can reach the TFTP server.)

3. Enter the following command to copy the software upgrade onto the PCMCIA flash card in the Control Module:

   system image add *<IPaddr-of-TFTP-host> <image-file-name>*
   Here is an example:

```
gs/r# system image add 10.50.11.12 gsr1010
Downloading image 'gsr10110' from host '10.50.11.12'
 to local image gsr1010 (takes about 3 minutes)
kernel: 100%
Image checksum validated.
Image added.
```

4. Enter the following command to list the images on the PCMCIA flash card and verify that the new image is on the card:

   ```
   system image list
   ```

   Here is an example:

   ```
   gs/r# system image list
   Images currently available:
    gsr1010
   ```

5. Enter the following command to select the image file the GIGAswitch/Router will use the next time you reboot the switch.

   ```
   system image choose <file-name>
   ```

   Here is an example:

   ```
   gs/r# system image choose gsr1010
   Making image gsr1010 the active image for next reboot
   ```

6. Enter the **system image list** command to verify the change.

Note: You do not need to activate this change.

Loading Boot PROM Software

The GIGAswitch/Router boots using the boot PROM software installed on the Control Module's internal memory. To upgrade the boot PROM software and boot using the upgraded image, use the following procedure.

1. Display the current boot settings by entering the following command:

   ```
   system show version
   ```

   Here is an example:

```
gs/r# system show version
Software Information
Software Information
  Software Version   : 1.0.1
  Copyright          : Copyright (c) 1996-1998 Cabletron Systems, Inc.
  Image Information  : Version 1.0.1, built on Wed Jun 10 22:49:07 1998
  Image Boot Location: file:/pc-flash/boot/gsr1010/
  Boot Prom Version  : prom-1.0.1
```

Note: In this example, the location “pc-flash” indicates that the GIGAswitch/Router is set to use the factory-installed software on the flash card.

2. Copy the software upgrade you want to install onto a TFTP server that the GIGAswitch/Router can access. (Use the **`ping`** command to verify that the GIGAswitch/Router can reach the TFTP server.)
3. Enter the following command to copy the boot PROM upgrade onto the internal memory in the Control Module:

   `system promimage upgrade` *`<IPaddr-of-TFTP-host> <image-file-name>`*

   Here is an example:

   ```
   gs/r# system promimage upgrade 10.50.11.12 prom2
   Downloading image 'prom2' from host '10.50.11.12'
    to local image prom2 (takes about 3 minutes)
   kernel: 100%
   Image checksum validated.
   Image added.
   ```

4. Enter the following command to verify that the new boot PROM software is on the internal memory of the Control Module:

   `system show version`

Activating Configuration Commands in the Scratchpad

The configuration commands you have entered using procedures in this chapter are in the Scratchpad but have not yet been activated. Use the following procedure to activate the configuration commands in the scratchpad.

1. If you have not already done so, enter the **`enable`** command to enter Enable mode in the CLI.
2. If you have not already done so, enter the **`configure`** command to enter Configure mode in the CLI.
3. Enter the following command:

   `save active`

   The CLI displays the following message:

   ```
   Do you want to make the changes Active? [y]
   ```

4. Enter **`yes`** to activate the changes.

Note: If you exit Configure mode (by entering the exit command or pressing Ctrl-z), the CLI will ask you whether you want to make the changes in the scratchpad active.

Copying Configuration to the Startup Configuration File

After you save the configuration commands in the scratchpad, the Control Module executes the commands and makes the corresponding configuration changes to the GIGAswitch/Router. However, if you power down or reboot the GIGAswitch/Router, the new changes are lost. Use the following procedure to save the changes into the Startup configuration file so that the GIGAswitch/Router reinstates the changes when you reboot the software.

1. If you have not already done so, enter the **`enable`** command to enter Enable mode in the CLI.
2. Enter the following command to copy the configuration changes in the Active configuration to the Startup configuration:

   ```
   copy active to startup
   ```

3. When the CLI displays the following message, enter **`yes`** to save the changes.

```
Are you sure you want to overwrite the Startup configuration? [n]
```

Note: You also can save active changes to the Startup configuration file from within Configure mode by entering the following command:

```
save startup
```

The new configuration changes are added to the Startup configuration file stored in the Control Module's boot flash.

Managing the GIGAswitch/Router

The GIGAswitch/Router contains numerous system facilities for system management. You can perform configuration management tasks on the GIGAswitch/Router including:

- Setting the GIGAswitch/Router name
- Setting the GIGAswitch/Router date and time
- Configuring the CLI
- Configuring SNMP services

Setting the GIGAswitch/Router Name

The GIGAswitch/Router name is set to **gs/r** by default. You may customize the name for the GIGAswitch/Router by performing the following in Configure mode:.

Set the GSR name.	**system set name** *<system-name>*

Setting the GIGAswitch/Router Date and Time

The GIGAswitch/Router system time keeps track of time as entered by the user. No time coordination is maintained between the GIGAswitch/Router and a source for Universal Time. To configure the GIGAswitch/Router date and time, enter the following command in Enable mode:

Set GSR date and time.	**system set date year** *<year>* **month** *<month>* **day** *<day>* **hour** *<hour>* **min** *<min>* **second** *<sec>*

Configuring the GIGAswitch/Router CLI

You can customize the CLI display format to a desired line length or row count. To configure the CLI terminal display, enter the following command in Enable mode:

Configure the CLI terminal display.	**cli set terminal rows** *<num>* **columns** *<num>*

Configuring SNMP Services

The GIGAswitch/Router accepts SNMP sets and gets from an SNMP manager. You can configure GIGAswitch/Router SNMP parameters including community strings and trap server target addresses.

To configure the GIGAswitch/Router SNMP community string, enter the following command in Configure mode:

Configure the SNMP community string.	**snmp set community** *<community-name>* **privilege read|read-write**

To configure the SNMP trap server target address, enter the following command in Configure mode:

Configure the SNMP trap server target address.	**snmp set target** *<IP-addr>* **community** *<community-name>* **[status enable|disable]**

Configuring DNS

The GIGAswitch/Router allows you to configure up to three Domain Name Service (DNS) servers.

To configure the DNS, the following command in Configure mode.

Configure DNS.	**system set dns server** *<IPaddr>***[,***<IPaddr>***[,***<IPaddr>***]] domain** *<name>*

Configuring HTTP Services

The GIGAswitch/Router contains an HTTP server for responding to access from CoreWatch. You have the ability to stop the HTTP server or disable authentication.

To configure the HTTP parameters, enter one of the following commands in Configure mode:

Stop the HTTP server.	**http stop**
Stop HTTP authentication.	**http disable authentication**

Monitoring Configuration

The GIGAswitch/Router provides many commands for displaying configuration information. After you add configuration items and commit them to the active configuration, you can display them using the following commands.

Display history buffer.	**cli show history**
Show terminal settings.	**cli show terminal**
Show accesses to the HTTP server.	**http show access**
Show all HTTP related information.	**http show all**
Show HTTP server status.	**http show server**
Show HTTP server related statistics.	**http show statistics**
Show all accesses to the SNMP agent.	**snmp show access**
Show all SNMP information.	**snmp show all**

Show chassis ID.	**`snmp show chassis-id`**
Show the SNMP community strings.	**`snmp show community`**
Show SNMP related statistics.	**`snmp show statistics`**
Show trap target related configuration.	**`snmp show trap`**
Show the active configuration of the system.	**`system show active-config`**
Show the contents of the boot log file, which contains all the system messages generated during bootup.	**`system show bootlog`**
Show the most recent Syslog messages kept in the local syslog message buffer.	**`system show syslog buffer`**
Show the contact information (administrator name, phone number, and so on).	**`system show contact`**
Show the GIGAswitch/Router date and time.	**`system show date`**
Show the IP addresses and domain names for DNS servers.	**`system show dns`**
Show GIGAswitch/Router hardware information.	**`system show hardware`**
Show GIGAswitch/Router location.	**`system show location`**
Show GIGAswitch/Router name.	**`system show name`**
Show the type of Power-On Self Test (POST) that should be performed.	**`system show poweron-selftest-mode`**
Show the configuration changes in the scratchpad. These changes have not yet been activated.	**`system show scratchpad`**
Show the startup configuration for the next reboot.	**`system show startup-config`**

Show the IP address of the SYSLOG server and the level of messages the GIGAswitch/Router sends to the server.	**`system show syslog`**
Lists the last five Telnet connections to the GIGAswitch/Router.	**`system show telnet-access`**
Show the default terminal settings (number of rows, number of columns, and baud rate).	**`system show terminal`**
Show GIGAswitch/Router uptime.	**`system show uptime`**
Show the software version running on the GIGAswitch/Router.	**`system show version`**

Chapter 2 Bridging Configuration Guide

Bridging Overview

The DIGITAL GIGAswitch/Router provides the following bridging functions:

- Complies with the IEEE 802.1d standard
- Complies with the IGMP multicast bridging standard
- Provides wire-speed address-based bridging or flow-based bridging
- Provides the ability to logically segment a transparently bridged network into virtual local-area networks (VLANs) based on physical ports or protocol (IP or IPX or bridged protocols like Appletalk)
- Allows frame filtering based on MAC address for bridged and multicast traffic
- Provides integrated routing and bridging, which supports bridging of intra-VLAN traffic and routing of inter-VLAN traffic

Spanning Tree (IEEE 802.1d)

Spanning tree (IEEE 802.1d) allows bridges to dynamically discover a subset of the topology that is loop-free. In addition, the loop-free tree that is discovered contains paths to every LAN segment.

Bridging Modes (Flow-Based and Address-Based)

The GIGAswitch/Router provides the following types of wire-speed bridging:

Address-based bridging – The GIGAswitch/Router performs this type of bridging by looking up the destination address in an L2 lookup table on the line card that receives the bridge packet from the network. The L2 lookup table indicates the exit port(s) for the bridged packet. If the packet is addressed to the GIGAswitch/Router's own MAC address, the packet is routed rather than bridged.

Flow-based bridging – The GIGAswitch/Router performs this type of bridging by looking up an entry in the L2 lookup table containing both the source and destination addresses of the received packet in order to determine how the packet is to be handled.

The GIGAswitch/Router ports perform address-based bridging by default but can be configured to perform flow-based bridging instead, on a per-port basis. A port cannot be configured to perform both types of bridging at the same time.

The GIGAswitch/Router performance is equivalent when performing flow-based bridging or address-based bridging. However, address-based bridging is more efficient because it requires fewer table entries while flow-based bridging provides tighter management and control over bridged traffic.

VLAN Overview

Virtual LANs (VLANs) are a means of dividing a physical network into several logical (virtual) LANs. The division can be done on the basis of various criteria, giving rise to different types of VLANs. For example, the simplest type of VLANs is the port-based VLAN. Port-based VLANs divide a network into a number of VLANs by assigning a VLAN to each port of a switching device. Then, any traffic received on a given port of a switch *belongs* to the VLAN associated with that port.

The primary use of VLANs is for broadcast containment. A layer-2 (L2) broadcast frame is normally transmitted all over a bridged network. By dividing the network into VLANs, the *range* of a broadcast is limited, i.e., the broadcast frame is transmitted only to the VLAN to which it belongs. This reduces the broadcast traffic on a network by an appreciable factor.

The type of VLAN depends upon one criterion: how a received frame is classified as belonging to a particular VLAN. VLANs can be categorized into the following types:

1. Port based
2. MAC address based
3. Protocol based
4. Subnet based
5. Multicast based
6. Policy based

Detailed information about these types of VLANs is beyond the scope of this manual. Each type of VLAN is briefly explained in the following subsections.

Port-based VLANs

Ports of L2 devices (switches, bridges) are assigned to VLANs. Any traffic received by a port is classified as belonging to the VLAN to which the port belongs. For example, if ports 1, 2, and 3 belong to the VLAN named “Marketing”, then a broadcast frame received by port 1 is transmitted on ports 2 and 3. It is not transmitted on any other port.

MAC-address-based VLANs

In this type of VLAN, each switch (or a central VLAN information server) keeps track of all MAC addresses in a network and maps them to VLANs, based on information configured by the network administrator. When a frame is received at a port, its destination MAC address is looked up in the VLAN database, which returns the VLAN to which this frame belongs.

This type of VLAN is powerful in the sense that network devices such as printers and workstations can be moved anywhere in the network without the need for network reconfiguration. However, the administration is intensive because all MAC addresses on the network need to be known and configured.

Protocol-based VLANs

Protocol-based VLANs divide the physical network into logical VLANs based on protocol. When a frame is received at a port, its VLAN is determined by the protocol of the packet. For example, there could be separate VLANs for IP, IPX and Appletalk. An IP broadcast frame will only be sent to all ports in the IP VLAN.

Subnet-based VLANs

Subnet-based VLANs are a subset of protocol based VLANs and determine the VLAN of a frame based on the subnet to which the frame belongs. To do this, the switch must look into the network layer header of the incoming frame. This type of VLAN behaves similar to a router by segregating different subnets into different broadcast domains.

Multicast-based VLANs

Multicast-based VLANs are created dynamically for multicast groups. Typically, each multicast group corresponds to a different VLAN. This ensures that multicast frames are received only by those ports that are connected to members of the appropriate multicast group.

Policy-based VLANs

Policy-based VLANs are the most general definition of VLANs. Each incoming (untagged) frame is looked up in a policy database, which determines the VLAN to which the frame belongs. For example, you could set up a policy which creates a special VLAN for all email traffic between the management officers of a company, so that this traffic will not be seen anywhere else.

GIGAswitch/Router VLAN Support

The GIGAswitch/Router supports:

- Port-based VLANs
- Protocol-based VLANs
- Subnet-based VLANs

When using the GIGAswitch/Router as an L2 bridge/switch, use the port-based and protocol-based VLAN types. When using the GIGAswitch/Router as a combined switch and router, use the subnet-based VLANs in addition to port-based and protocol-based VLANs. It is not necessary to remember the types of VLANs in order to configure the GIGAswitch/Router, as seen in the section on configuring the GIGAswitch/Router.

VLANs and the GIGAswitch/Router

VLANs are an integral part of the GIGAswitch/Router. The GIGAswitch/Router switching router can function as a layer-2 (L2) switch as well as fully-functonal layer-3 (L3) router. Hence it can be viewed as a switch and a router in one box. To provide maximum performance and functionality, the L2 and L3 aspects of the GIGAswitch/Router switching router is tightly coupled.

The GIGAswitch/Router can be used purely as an L2 switch. Frames arriving at any port are bridged and not routed. In this case, setting up VLANs and associating ports with VLANs is all that is required. You can set up the GIGAswitch/Router switching router to use port-based VLANs, protocol-based VLANs, or a mixture of the two types.

The GIGAswitch/Router can also be used purely as a router, i.e., each physical port of the GIGAswitch/Router is a separate routing interface. Packets received at any interface are routed and not bridged. In this case, no VLAN configuration is required. Note that VLANs are still created implicitly by the GIGAswitch/Router as a result of creating L3 interfaces for IP and/or IPX. However, these implicit VLANs do not need to be created or configured manually. The implicit VLANs created by the GIGAswitch/Router are subnet-based VLANs.

Most commonly, an GIGAswitch/Router is used as a combined switch and router. For example, it may be connected to two subnets S1 and S2. Ports 1-8 belong to S1 and ports 9-16 belong to S2. The required behavior of the GIGAswitch/Router is that intra-subnet frames be bridged and inter-subnet packets be routed. In other words, traffic between two workstations that belong to the same subnet should be bridged, and traffic between two workstations that belong to different subnets should be routed.

The GIGAswitch/Router switching routers use VLANs to achieve this behavior. This means that a L3 subnet (i.e., an IP or IPX subnet) is mapped to a VLAN. A given subnet maps to exactly one and only one VLAN. With this definition, the terms *VLAN* and *subnet* are almost interchangeable.

To configure an GIGAswitch/Router as a combined switch and router, the administrator must create VLANs whenever multiple ports of the GIGAswitch/Router are to belong to a particular VLAN/subnet. Then the VLAN must be *bound to* an L3 (IP/IPX) interface so that the GIGAswitch/Router knows which VLAN maps to which IP/IPX subnet.

Ports, VLANs, and L3 Interfaces

The term *port* refers to a physical connector on the GIGAswitch/Router, such as an ethernet port. Each port must belong to at least one VLAN. When the GIGAswitch/Router is unconfigured, each port belongs to a VLAN called the “default VLAN”. By creating VLANs and adding ports to the created VLANs, the ports are moved from the default VLAN to the newly created VLANs.

Unlike traditional routers, the GIGAswitch/Router has the concept of logical interfaces rather than physical interfaces. An L3 interface is a logical entity created by the administrator. It can contain more than one physical port. When an L3 interface contains exactly one physical port, it is equivalent to an interface on a traditional router. When an L3 interface contains several ports, it is equivalent to an interface of a traditional router which is connected to a layer-2 device such as a switch or bridge.

Access Ports and Trunk Ports (802.1Q support)

The ports of an GIGAswitch/Router can be classified into two types, based on VLAN functionality: **access ports** and **trunk ports**. By default, a port is an access port. An access port can belong to at most one VLAN of the following types: IP, IPX or bridged protocols. The GIGAswitch/Router can automatically determine whether a received frame is an IP frame, an IPX frame or neither. Based on this, it selects a VLAN for the frame. Frames transmitted out of an access port are *untagged*, meaning that they contain no special information about the VLAN to which they belong. Untagged frames are classified as belonging to a particular VLAN based on the protocol of the frame and the VLAN configured on the receiving port for that protocol.

For example, if port 1 belongs to VLAN *IPX_VLAN* for IPX, VLAN *IP_VLAN* for IP and VLAN *OTHER_VLAN* for any other protocol, then an IP frame received by port 1 is classified as belonging to VLAN *IP_VLAN*.

Trunk ports (802.1Q) are usually used to connect one VLAN-aware switch to another. They carry traffic belonging to several VLANs. For example, suppose that GIGAswitch/Router A and B are both configured with VLANs V1 and V2.

Then a frame arriving at a port on GIGAswitch/Router A must be sent to GIGAswitch/Router B, if the frame belongs to VLAN V1 or to VLAN V2. Thus the ports on GIGAswitch/Router A and B which connect the two GIGAswitch/Routers together must belong to both VLAN V1 and VLAN V2. Also, when these ports receive a frame, they must be able to determine whether the frame belongs to V1 or to V2. This is accomplished by “tagging” the frames, i.e., by prepending information to the frame in order to identify the VLAN to which the frame belongs. In the GIGAswitch/Router switching routers, trunk ports always transmit and receive tagged frames only. The format of the tag is specified by the IEEE 802.1Q standard. The only exception to this is Spanning Tree Protocol frames, which are transmitted as untagged frames.

Explicit and Implicit VLANs

As mentioned earlier, VLANs can either be created explicitly by the administrator (explicit VLANs) or are created implicitly by the GIGAswitch/Router when L3 interfaces are created (implicit VLANs).

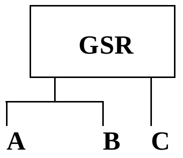
Configuring GIGAswitch/Router Bridging Functions

Configuring Address-based or Flow-based Bridging

The GIGAswitch/Router ports perform address-based bridging by default but can be configured to perform flow-based bridging instead of address-based bridging, on a per-port basis. A port cannot be configured to perform both types of bridging at the same time.

The GIGAswitch/Router performance is equivalent when performing flow-based bridging or address-based bridging. However, address-based bridging is more efficient because it requires fewer table entries while flow-based bridging provides tighter management and control over bridged traffic.

For example, the following illustration shows an GIGAswitch/Router (GSR) with traffic being sent from port A to port B, port B to port A, port B to port C, and port A to port C.



The corresponding bridge tables for address-based and flow-based bridging are shown below. As shown, the bridge table contains more information on the traffic patterns when flow-based bridging enabled compared to address-based bridging.

Address-Based Bridge Table	Flow-Based Bridge Table
A (source)	A ’ B
B (source)	B ’ A

Address-Based Bridge Table	Flow-Based Bridge Table
C (destination)	B ’ C
	A ’ C

With the GIGAswitch/Router configured in flow-based bridging mode, the network manager has “per flow” control of layer-2 traffic. The network manager can then apply Quality of Service (QoS) policies or security filters based layer-2 traffic flows.

To enable a port to flow-based bridging, enter the following command in Configure Mode.

Configure a port for flow-based bridging.	**port flow-bridging** *<port-list>***|all-ports**

To change a port from flow-based bridging to address-based bridging, enter the following command in Configure mode:

Change a port from flow-based bridging to address-based bridging.	**negate** *<line-number of active config containing command>***: port flow-bridging** *<port-list>***|all-ports**

Configuring Spanning Tree

The GIGAswitch/Router supports only one spanning tree process per GIGAswitch/ Router. By default, spanning tree is disabled on the GIGAswitch/Router. To enable spanning tree on the GIGAswitch/Router, you perform the following task on the ports where you want spanning tree enabled.

Note: If you are running spanning tree on one or more VLANs, you must enable spanning tree on all ports belonging to each VLAN.

Enable spanning tree on one or more ports.	**stp enable port** *<port-list>*

Adjusting Spanning-Tree Parameters

You may need to adjust certain spanning-tree parameters if the default values are not suitable for your bridge configuration. Parameters affecting the entire spanning tree are configured with variations of the bridge global configuration command. Interface-specific parameters are configured with variations of the bridge-group interface configuration command.

You can adjust spanning-tree parameters by performing any of the tasks in the following sections:

- Set the Bridge Priority
- Set an Interface Priority

Note: Only network administrators with a good understanding of how bridges and the Spanning-Tree Protocol work should make adjustments to spanning-tree parameters. Poorly chosen adjustments to these parameters can have a negative impact on performance. A good source on bridging is the IEEE 802.1d specification.

Setting the Bridge Priority

You can globally configure the priority of an individual bridge when two bridges tie for position as the root bridge, or you can configure the likelihood that a bridge will be selected as the root bridge. The lower the bridge's priority, the more likely the bridge will be selected as the root bridge. This priority is determined by default; however, you can change it.

To set the bridge priority, enter the following command in Configure mode:

Set the bridge priority.	**`stp set bridging priority`** *`<num>`*

Setting a Port Priority

You can set a priority for an interface. When two bridges tie for position as the root bridge, you configure an interface priority to break the tie. The bridge with the lowest interface value is elected.

To set an interface priority, enter the following command in Configure mode:

Establish a priority for a specified interface.	**`stp set port`** *`<port-list>`* **`priority`** *`<num>`*

Assigning Port Costs

Each interface has a port cost associated with it. By convention, the port cost is 1000/data rate of the attached LAN, in Mbps. You can set different port costs.

To assign port costs, enter the following command in Configure mode:

Set a different port cost other than the defaults.	**`stp set port`** *`<port-list>`* **`port-cost`** *`<num>`*

Adjusting Bridge Protocol Data Unit (BPDU) Intervals

You can adjust BPDU intervals as described in the following sections:

- Adjust the Interval between Hello BPDUs
- Define the Forward Delay Interval
- Define the Maximum Idle Interval

Adjusting the Interval Between Hello Times

You can specify the interval between hello time.

To adjust this interval, enter the following command in Configure mode:

Specify the interval between hello time	`stp set bridging hello-time` *<num>*

Defining the Forward Delay Interval

The forward delay interval is the amount of time spent listening for topology change information after an interface has been activated for bridging and before forwarding actually begins.

To change the default interval setting, enter the following command in Configure mode:

Set the default of the forward delay interval.	`stp set bridging forward-delay` *<num>*

Defining the Maximum Age

If a bridge does not hear BPDUs from the root bridge within a specified interval, it assumes that the network has changed and recomputes the spanning-tree topology.

To change the default interval setting, enter the following command in Configure mode:

Change the amount of time a bridge will wait to hear BPDUs from the root bridge.	`stp set bridging max-age` *<num>*

Configuring a Port or Protocol based VLAN

To create a port or protocol based VLAN, perform the following steps in the Configure mode.

1. Create a port or protocol based VLAN
2. Add physical ports to a VLAN

Creating a Port or Protocol Based VLAN

To create a VLAN, perform the following command in the Configure mode.

Create a VLAN.	**`vlan create`** *`<vlan-name> <type>`* **`id`** *`<num>`*

Adding Ports to a VLAN

To add ports to a VLAN, perform the following command in the Configure mode.

Add ports to a VLAN.	**`vlan add ports`** *`<port-list>`* **`to`** *`<vlan-name>`*

Configuring VLAN Trunk Ports

The GIGAswitch/Router supports standards-based VLAN trunking between multiple GIGAswitch/Routers as defined by IEEE 802.1Q. 802.1Q adds a header to a standard Ethernet frame which includes a unique VLAN id per trunk between two GIGAswitch/ Routers. These VLAN ids extend the VLAN broadcast domain to more than one GIGAswitch/Router.

To configure a VLAN trunk, perform the following command in the Configure mode.

Configure 802.1Q VLAN trunks.	**`vlan make`** *`<port-type> <port-list>`*

Configuring Bridging for Non-IP/IPX Protocols

By default, all non-routable protocols (AppleTalk and DECnet) are bridged within the GIGAswitch/Router. All physical ports containing non-routable protocols should be assigned to the same VLAN, thus allowing bridging between ports. Routing can still be performed on the defined VLAN by assigning an IP or IPX interface.

Configuring Layer-2 Filters

Layer-2 security filters on the GIGAswitch/Router allow you to configure ports to filter specific MAC addresses. When defining a Layer-2 security filter, you specify to which ports you want the filter to apply. Refer to the *“Security Configuration Chapter”* for details on configuring Layer-2 filters. You can specify the following security filters:

- Address filters

  These filters block traffic based on the frame's source MAC address, destination MAC address, or both source and destination MAC addresses in flow bridging mode. Address filters are always configured and applied to the input port.

- Port-to-address lock filters

  These filters prohibit a user connected to a locked port or set of ports from using another port.

- Static entry filters

  These filters allow or force traffic to go to a set of destination ports based on a frame's source MAC address, destination MAC address, or both source and destination MAC addresses in flow bridging mode. Static entries are always configured and applied at the input port.

- Secure port filters

  A secure filter shuts down access to the GIGAswitch/Router based on MAC addresses. All packets received by a port are dropped. When combined with static entries, however, these filters can be used to drop all received traffic but allow some frames to go through.

Monitoring Bridging

The GIGAswitch/Router provides display of bridging statistics and configurations contained in the GIGAswitch/Router.

To display bridging information, enter the following commands in Enable mode.

Show IP routing table.	**ip show routes**
Show all MAC addresses currently in the l2 tables.	**l2-tables show all-macs**
Show l2 table information on a specific port.	**l2-tables show port-macs**
Show information the master MAC table.	**l2-tables show mac-table-stats**
Show information on a specific MAC address.	**l2-tables show mac**
Show information on MACs registered.	**l2-table show bridge-management**
Show all VLANs.	**vlan list**

Configuration Examples

Creating an IP or IPX VLAN

VLANs are used to associate physical ports on the GIGAswitch/Router with connected hosts that may be physically separated but need to participate in the same broadcast domain. To associate ports to a VLAN, you must first create an IP or IPX VLAN and then assign ports to the VLAN.

For example, servers connected to port gi.1.(1-2) on the GIGAswitch/Router need to communicate with clients connected to et.4.(1-8). You can associate all the ports containing the clients and servers to an IP VLAN called ‘BLUE’.

Step 1: Create an IP VLAN named ‘BLUE’

```
gs/r(config)# vlan create BLUE ip
```

Step 2: Assign ports to the ‘BLUE’ VLAN.

```
gs/r(config)# vlan add ports et.1.(1-8),gi.1.(1-2) to BLUE
```

Chapter 3 IP Routing Configuration Guide

This chapter describes how to configure IP interfaces and general non-protocol-specific routing parameters.

IP Routing Overview

Internet Protocol (IP) is a packet-based protocol used to exchange data over computer networks. IP handles addressing, routing, fragmentation, reassembly, and protocol demultiplexing. In addition, IP specifies how hosts and routers should process packets, handle errors and discard packets. IP forms the foundation upon which transport layer protocols, such as TCP or UDP, interoperate over a routed network.

The Transmission Control Protocol (TCP) is built upon the IP layer. TCP is a connection-oriented protocol that specifies the data format, buffering and acknowledgments used in the transfer of data. TCP is a full-duplex connection which also specifies the procedures that the computers use to ensure that the data arrives correctly.

The User Datagram Protocol (UDP) provides the primary mechanism that applications use to send datagrams to other application programs. UDP is a connectionless protocol that does not guarantee delivery of datagrams between applications. Applications which use UDP are responsible for ensuring successful data transfer by employing error handling, retransmission and sequencing techniques.

TCP and UDP also specify “ports,” which identify the application which is using TCP/UDP. For example, a web server would typically use TCP/UDP port 80, which specifies HTTP-type traffic.

The GIGAswitch/Router supports standards based TCP, UDP, and IP.

IP Routing Protocols

The GIGAswitch/Router supports standards based unicast and multicast routing. Unicast routing protocol support include Interior Gateway Protocols and Exterior Gateway Protocols. Multicast routing protocols are used to determine how multicast data is transferred in a routed environment.

Unicast Routing Protocols

Interior Gateway Protocols are used for routing networks that are within an “autonomous system,” a network of relatively limited size. All IP interior gateway protocols must be specified with a list of associated networks before routing activities can begin. A routing process listens to updates from other routers on these networks and broadcasts its own routing information on those same networks. The GIGAswitch/Router supports the following Interior Gateway Protocols:

- Routing Information Protocol (RIP) Version 1, 2 (RFC 1058, 1723)
- Open Shortest Path First (OSPF) Version 2 (RFC 1583)

Exterior Gateway Protocols are used to transfer information between different “autonomous systems”. The GIGAswitch/Router supports the following Exterior Gateway Protocol:

- Border Gateway Protocol (BGP) Version 3, 4 (RFC 1267, 1771)

Multicast Routing Protocols

IP multicasting allows a host to send traffic to a subset of all hosts. These hosts subscribe to group membership, thus notifying the GIGAswitch/Router of participation in a multicast transmission.

Multicast routing protocols are used to determine which routers have directly attached hosts, as specified by IGMP, that have membership to a multicast session. Once host memberships are determined, routers use multicast routing protocols, such as DVMRP, to forward multicast traffic between routers.

The GIGAswitch/Router supports the following multicast routing protocols:

- Distance Vector Multicast Routing Protocol (DVMRP) RFC 1075
- Internet Group Management Protocol (IGMP) as described in RFC 2236

The GIGAswitch/Router also supports the latest DVMRP Version 3.0 draft specification, which includes mtrace, Generation ID and Pruning/Grafting.

Configuring IP Interfaces and Parameters

This section provides an overview of configuring various IP parameters and setting up IP interfaces.

Configuring IP Addresses to Ports

You can configure one IP interface directly to physical ports. Each port can be assigned multiple IP addresses representing multiple subnets connected to the physical port.

To configure an IP interface to a port, enter one of the following commands in Configure mode.

Configure an IP interface to a physical port.	**interface create ip** *<InterfaceName>* **address-mask** *<ipAddr-mask>* **port** *<port>*

Configure a secondary address to an existing IP interface.	**interface add ip** *<InterfaceName>* **address-netmask** *<ipAddr-mask>* **[broadcast** *<ipaddr>***]**

Configuring IP Interfaces for a VLAN

You can configure one IP interface per VLAN. Once an IP interface has been assigned to a VLAN, you can add a secondary IP addresses to the VLAN.

To configure a VLAN with an IP interface, enter the following command in Configure mode:

Create an IP interface for a VLAN.	**interface create ip** *<InterfaceName>* **address-mask** *<ipAddr-mask>* **vlan** *<name>*
Configure a secondary address to an existing VLAN.	**interface add ip** *<InterfaceName>* **address-netmask** *<ipAddr-mask>* **vlan** *<name>*

Specifying Ethernet Encapsulation Method

The DIGITAL GIGAswitch/Router supports two encapsulation types for IP. You can configure encapsulation type on a per interface basis.

- Ethernet II: The standard ARPA Ethernet Version 2.0 encapsulation, which uses a 16-bit protocol type code (the default encapsulation method)
- 802.3 SNAP: SNAP IEEE 802.3 encapsulation, in which the type code becomes the frame length for the IEEE 802.2 LLC encapsulation (destination and source Service Access Points, and a control byte)

To configure IP encapsulation, enter one of the following commands in Configure mode.

Configure Ethernet II encapsulation.	**interface create ip** *<InterfaceName>* **output-mac-encapsulation ethernet_II**
Configure 802.3 SNAP encapsulation.	**interface create ip** *<InterfaceName>* **output-mac-encapsulation ethernet_snap**

Configuring Address Resolution Protocol

The GIGAswitch/Router allows you to configure Address Resolution Protocol (ARP) table entries and parameters. ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated MAC address. Once a media or MAC address is determined, the IP address/media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network.

Configuring ARP Cache Entries

You can add and delete entries in the ARP cache. To add or delete static ARP entries, enter one of the the following commands in Configure mode:

Add a static ARP entry.	**arp add** *<host>* **mac-addr** *<MAC-addr>* **exit-port** *<port>*
Clear a static ARP entry.	**arp clear** *<host>*

Configuring Proxy ARP

The GIGAswitch/Router can be configured for proxy ARP. The GIGAswitch/Router uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the MAC address of hosts on other networks or subnets. Through Proxy ARP, the GIGAswitch/Router will respond to ARP requests from a host with a ARP reply packet containing the GIGAswitch/Router MAC address. Proxy ARP is enabled by default on the GIGAswitch/Router.

To disable proxy ARP, enter the following command in Configure mode:

Disable Proxy ARP on an interface.	**ip disable-proxy-arp interface** *<Interface-Name>***|all**

Configuring DNS Parameters

The GIGAswitch/Router can be configured to specify DNS servers which supply name services for DNS requests. You can specify up to three DNS servers.

To configure DNS servers, enter the following command in Configure mode:

Configure a DNS server.	**system set dns server** *<IPaddr>***[,***<IPaddr>***[,***<IPaddr>***]]**

You can also specify a domain name for the GIGAswitch/Router. The domain name is used by the GIGAswitch/Router to respond to DNS requests.

To configure a domain name, enter the following command in Configure mode:

Configure a domain name.	**system set dns domain** *<name>*

Configuring IP Services (ICMP)

The GIGAswitch/Router provides ICMP message capabilities including ping and traceroute. Ping allows you to determine the reachability of a certain IP host. Traceroute allows you to trace the IP gateways to an IP host.

To access ping or traceroute on the GIGAswitch/Router, enter the following commands in Enable mode:

Specify ping.	**ping** *<hostname-or-IPaddr>* **packets** *<num>* **size** *<num>* **wait** *<num>* **[flood] [dontroute]**
Specify traceroute.	**traceroute** *<host>* **[max-ttl** *<num>***] [probes** *<num>***] [size** *<num>***] [source** *<secs>***] [tos** *<num>***] [wait-time** *<secs>***] [verbose] [noroute]**

Monitoring IP Parameters

The GIGAswitch/Router provides display of IP statistics and configurations contained in the routing table. Information displayed provides routing and performance information.

To display IP information, enter the following command in Enable mode:

Show ARP table entries.	**arp show entries**
Show ARP table settings.	**arp show settings**
Show IP interface configuration	**interface show ip**
Show all TCP/UDP connections and services.	**ip show connections [no-lookup]**
Show configuration of IP interfaces.	**ip show interfaces [***<interface-name>***]**
Show IP routing table information.	**ip show routes**
Show ARP entries in routing table.	**ip show routes show-arps**
Show DNS parameters.	**system show dns**

Configuration Examples

Assigning IP/IPX Interfaces

To enable routing on the GIGAswitch/Router, you must assign an IP or IPX interface to a VLAN. To assign an IP or IPX interface named ‘RED’ to the ‘BLUE’ VLAN, perform the following:

```
gs/r(config)# interface create ip RED address-netmask 10.50.0.1/
255.255.0.0 vlan BLUE
```

You can also assign an IP or IPX interface directly to a physical port. For example, to assign an IP interface ‘RED’ to physical port et.3.4, perform the following:

```
gs/r(config)# interface create ip RED address-netmask 10.50.0.0/
255.255.0.0 port et.3.4
```

Chapter 4 RIP Configuration Guide

RIP Overview

This chapter describes how to configure Routing Information Protocol (RIP) in the DIGITAL GIGAswitch/Router. RIP is a distance-vector routing protocol for use in small networks. RIP is described in RFC 1723. A router running RIP broadcasts updates at set intervals. Each update contains paired values where each pair consists of an IP network address and an integer distance to that network. RIP uses a hop count metric to measure the distance to a destination.

The DIGITAL GIGAswitch/Router provides support for RIP Version 1 and 2. The GIGAswitch/Router implements plain text and MD5 authentication methods for RIP Version 2.

The protocol independent features that apply to RIP are described in the section "IP Routing Configuration Guide" on page 3 - 1.

Configuring RIP

By default, RIP is disabled on the GIGAswitch/Router and on each of the attached interfaces. To configure RIP on the GIGAswitch/Router, follow these steps:

1. Start the RIP process by using the *rip start* command.
2. Use the *rip add interface* command to inform RIP about the attached interfaces.

Enabling and Disabling RIP

To enable or disable RIP, enter one of the following commands in Configure mode.

Enable RIP.	**`rip start`**
Disable RIP.	**`rip stop`**

Configuring RIP Interfaces

To configure RIP in the GIGAswitch/Router, you must first add interfaces to inform RIP about attached interfaces.

To add RIP interfaces, enter the following commands in Configure mode.

Add interfaces to the RIP process.	**rip add interface** *<interfacename-or-IPaddr>*
Add gateways from which the GIGAswitch/Router will accept RIP updates.	**rip add trusted-gateway** *<interfacename-or-IPaddr>*
Define the list of routers to which RIP sends packets directly, not through multicast or broadcast.	**rip add source-gateway** *<interfacename-or-IPaddr>*

Configuring RIP Parameters

No further configuration is required and the system default parameters will be used by RIP to exchange routing information. These default parameters may be modified to suit your needs by using the *rip set interface* command.

RIP Parameter	Default Value
Version number	RIP v1
Check-zero for RIP reserved parameters	Enabled
Whether RIP packets should be broadcast	Choose
Preference for RIP routes	100
Metric for incoming routes	1
Metric for outgoing routes	0
Authentication	None
Update interval	30 seconds

To change RIP parameters, enter the following commands in Configure mode.

Set RIP Version on an interface to RIP V1.	**rip set interface** *<interfacename-or-IPaddr>***|all version 1**
Set RIP Version on an interface to RIP V2.	**rip set interface** *<interfacename-or-IPaddr>***|all version 2**
Specify that RIP V2 packets should be multicast on this interface.	**rip set interface** *<interfacename-or-IPaddr>***|all type multicast**
Specify that RIP V2 packets that are RIP V1-compatible should be broadcast on this interface.	**rip set interface** *<interfacename-or-IPaddr>***|all type broadcast**
Change the metric on incoming RIP routes.	**rip set interface** *<interfacename-or-IPaddr>***|all metric-in** *<num>*
Change the metric on outgoing RIP routes.	**rip set interface** *<interfacename-or-IPaddr>***|all metric-out** *<num>*
Set the authentication method to simple text up to 8 characters.	**rip set interface** *<interfacename-or-IPaddr>***|all authentication-method simple**
Set the authentication method to MD5.	**rip set interface** *<interfacename-or-IPaddr>***|all authentication-method md5**
Specify the metric to be used when advertising routes that were learned from other protocols.	**rip set default-metric** *<num>*

Configuring RIP Route Preference

You can set the preference of routes learned from RIP.

To configure RIP route preference, enter the following command in Configure mode.

Set the preference of routes learned from RIP.	**rip set preference** *<num>*

Configuring RIP Route Default-Metric

You can define the metric used when advertising routes via RIP that were learned from other protocols. The default value for this parameter is 16 (unreachable). To export routes from other protocols into RIP, you must explicitly specify a value for the default-metric parameter. The metric specified by the default-metric parameter may be overridden by a metric specified in the export command.

To configure default-metric, enter the following command in Configure mode.

Define the metric used when advertising routes via RIP that were learned from other protocols.	**rip set default-metric** *<num>*

For *<num>*, you must specify a number between 1 and 16.

Monitoring RIP

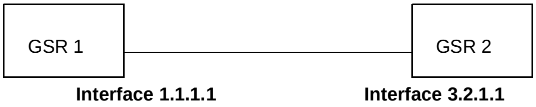
The *rip trace* command can be used to trace all rip request and response packets.

To monitor RIP information, enter the following commands in Enable mode.

Show all RIP information.	**rip show all**
Show RIP export policies.	**rip show export-policy**
Show RIP global information.	**rip show globals**
Show RIP import policies.	**rip show import-policy**
Show RIP information on the specified interface.	**rip show interface** *<Name or IP-addr>*
Show RIP interface policy information.	**rip show interface-policy**
Show detailed information of all RIP packets	**rip trace packets detail**
Show detailed information of all packets received by the router.	**rip trace packets receive**
Show detailed information of all packets sent by the router.	**rip trace packets send**
Show detailed information of all request received by the router.	**rip trace request receive**

Show detailed information of all response received by the router.	**`rip trace response receive`**
Show detailed information of response packets sent by the router.	**`rip trace response send`**
Show detailed information of request packets sent by the router.	**`rip trace send request`**
Show RIP timer information.	**`rip show timers`**

Configuration Example



```
! Example configuration
!
! Create interface gsr1-if1 with ip address 1.1.1.1/16 on port
et.1.1 on GSR-1
 interface create ip gsr1-if1 address-netmask 1.1.1.1/16 port
et.1.1
!
! Configure rip on GSR-1
rip add interface gsr1-if1
rip set interface gsr1-if1 version 2
rip start
!
!
! Set authentication method to md5
rip set interface gsr1-if1 authentication-method md5
!
!  Change default metric-in
rip set interface gsr1-if1 metric-in 2
!
! Change default metric-out
rip set interface gsr1-if1 metric-out 3
```

Chapter 5 OSPF Configuration Guide

OSPF Overview

Open Shortest Path First (OSPF) is a link-state routing protocol that supports IP subnetting and authentication. The GIGAswitch/Router supports OSPF Version 2.0 as defined in RFC 1583. Each link-state message contains all the links connected to the router with a specified cost associated with the link.

The GIGAswitch/Router supports the following OSPF functions:

- Stub Areas: Definition of stub areas is supported
- Authentication: Simple password and MD5 authentication methods are supported within an area
- Virtual Links: Virtual links are supported
- Route Redistribution: Routes learned via RIP, BGP, or any other sources can be redistributed into OSPF. OSPF routes can be redistributed into RIP or BGP
- Interface Parameters: Parameters that can be configured include interface output cost, retransmission interval, interface transmit delay, router priority, router dead and hello intervals, and authentication key

Configuring OSPF

To configure OSPF on the GIGAswitch/Router, you must enable OSPF, create OSPF areas, assign interfaces to OSPF areas, and, if necessary, specify any of the OSPF interface parameters.

To configure OSPF, you may need to perform some or all of the following tasks:

1. Enable OSPF.
2. Create OSPF areas.
3. Create an IP interface or assign an IP interface to a VLAN.
4. Add IP interfaces to OSPF areas.
5. Configure OSPF interface parameters, if necessary.

Note: By default, the priority of an OSPF router for an interface is set to zero, which makes the router ineligible from becoming a designated router on the network to which the interface belongs. To make the router eligible to become a designated router, you must set the priority to a non-zero value.

The default cost of an OSPF interface is 1. The cost of the interface should be inversely proportional to the bandwidth of the interface; if the GIGAswitch/Router has interfaces with differing bandwidths, the OSPF costs should be set accordingly.

6. Add IP networks to OSPF areas.
7. Create virtual links, if necessary.

Enabling OSPF

OSPF is disabled by default on the GIGAswitch/Router.

To enable or disable OSPF, enter one of the following commands in Configure mode.

Enable OSPF.	**`ospf start`**
Disable OSPF.	**`ospf stop`**

Configuring OSPF Interface Parameters

You can configure the OSPF interface parameters shown in the table below.

OSPF Parameter	**Default Value**
Interface OSPF State (Enable/Disable)	Enable (except for virtual links)
Cost	1
No multicast	Default is using multicast mechanism.
Retransmit interval	5 seconds
Transit delay	1 second
Priority	0
Hello interval	10 seconds (broadcast), 30 (non broadcast)
Router dead interval	4 times the hello interval
Poll Interval	120 seconds
Key chain	N/A
Authentication Method	None

To configure OSPF interface parameters, enter one of the following commands in Configure mode:

Enable OSPF state on interface.	**ospf set interface <***name-or-IPaddr***>|all state disable|enable**
Specify the cost of sending a packet on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all cost** *<num>*
Specify the priority for determining the designated router on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all priority** *<num>*
Specify the interval between OSPF hello packets on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all hello-interval** *<num>*
Configure the retransmission interval between link state advertisements for adjacencies belonging to an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all retransmit-interval** *<num>*
Specify the number of seconds required to transmit a link state update on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all transit-delay** *<num>*
Specify the time a neighbor router will listen for OSPF hello packets before declaring the router down.	**ospf set interface <***name-or-IPaddr***>|all router-dead-interval** *<num>*
Disable IP multicast for sending OSPF packets to neighbors on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all no-multicast**
Specify the poll interval on an OSPF interface.	**ospf set interface <***name-or-IPaddr***>|all poll-interval** *<num>*
Specify the identifier of the key chain containing the authentication keys.	**ospf set interface <***name-or-IPaddr***>|all key-chain** *<num-or-string>*
Specify the authentication method to be used on this interface.	**ospf set interface <***name-or-IPaddr***>|all authentication-method none|simple|md5**

Configuring an OSPF Area

OSPF areas are a collection of subnets that are grouped in a logical fashion. These areas communicate with other areas via the backbone area. Once OSPF areas are created, you can add interfaces, stub hosts, and summary ranges to the area.

In order to reduce the amount of routing information propagated between areas, you can configure summary-ranges on Area Border Routers (ABRs). On the GIGAswitch/Router, summary-ranges are created using the **ospf add network** command – the networks specified using this command describe the scope of an area. Intra-area Link State Advertisements (LSAs) that fall within the specified ranges are not advertised into other areas as inter-area routes. Instead, the specified ranges are advertised as summary network LSAs.

To create areas and assign interfaces, enter the following commands in the Configure mode.

Create an OSPF area.	**ospf create area** *<area-num>*|**backbone**
Add an interface to an OSPF area.	**ospf add interface <***name-or-IPaddr***> [to-area** *<area-addr>*|**backbone] [type broadcast|non-broadcast]**
Add a stub host to an OSPF area.	**ospf add stub-host [to-area** *<area-addr>*|**backbone] [cost** *<num>***]**
Add a network to an OSPF area for summarization.	**ospf add network** *<IPaddr/mask>* **[to-area** *<area-addr>*|**backbone] [restrict] [host-net]**

Configuring OSPF Area Parameters

The GIGAswitch/Router allows configuration of various OSPF area parameters, including stub areas, stub cost and authentication method. Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR, into the stub area for destinations outside the autonomous system. Stub cost specifies the cost to be used to inject a default route into a stub area. An authentication method for OSPF packets can be specified on a per-area basis.

To configure OSPF area parameters, enter the following commands in the Configure mode.

Specify an OSPF stub area.	**ospf set area** *<area-num>* **stub**
Specify the cost to be used to inject a default route into an area.	**ospf set area** *<area-num>* **stub-cost** *<num>*
Specify the authentication method to be used by neighboring OSPF routers.	**ospf set area** *<area-num>* **[stub] [authentication-method none|simple|md5]**

Creating Virtual Links

In OSPF, virtual links can be established:

- To connect an area via a transit area to the backbone
- To create a redundant backbone connection via another area

Each Area Border Router must be configured with the same virtual link. Note that virtual links cannot be configured through a stub area.

To configure virtual links, enter the following commands in the Configure mode.

Create a virtual link.	**ospf add virtual-link** *<number-or-string>* **[neighbor** *<IPaddr>***] [transit-area** *<area-num>***]**
Set virtual link parameters.	**ospf set virtual-link** *<number-or-string>* **[state disable|enable] [cost** *<num>***] [retransmit-interval** *<num>***] [transit-delay** *<num>***] [priority** *<num>***] [hello-interval** *<num>***] [router-dead-interval** *<num>***] [poll-interval** *<num>***]**

Configuring Autonomous System External (ASE) Link Advertisements

These parameters specify the defaults used when importing OSPF AS External (ASE) routes into the routing table and exporting routes from the routing table into OSPF ASEs.

To specify AS external link advertisements parameters, enter the following commands in the Configure mode:

Specify the interval which AS external link advertisements will be generated and flooded to an OSPF AS.	**ospf set export-interval** *<num>*
Specify the number of AS external link advertisements which will be generated and flooded to an OSPF AS.	**ospf set export-limit** *<num>*
Specify AS external link advertisement default parameters.	**ospf set ase-defaults [preference** *<num>***]** **[cost** *<num>***]** **[type** *<num>***] [inherit-metric]**

Configuring OSPF over Non-Broadcast Multiple Access

You can configure OSPF over NBMA circuits to limit the number of Link State Advertisements (LSAs). LSAs are limited to initial advertisements and any subsequent changes. Periodic LSAs over NBMA circuits are suppressed.

To configure OSPF over WAN circuits, enter the following command in Configure mode:

Configure OSPF over a WAN circuit.	**ospf add nbma-neighbor** *<hostname-or-IPaddr>* **to-interface <***name-or-IPaddr***> [eligible]**

Monitoring OSPF

The GIGAswitch/Router provides display of OSPF statistics and configurations contained in the routing table. Information displayed provides routing and performance information.

To display OSPF information, enter the following commands in Enable mode.

Show IP routing table.	**ip show table routing**
Monitor OSPF error conditions.	**ospf monitor errors destination** *<hostname-or-IPaddr>*
Show information on all interfaces configured for OSPF.	**ospf monitor interfaces destination** *<hostname-or-IPaddr>*
Display link state advertisement information.	**ospf monitor lsa destination** *<hostname-or-IPaddr>*
Display the link state database.	**ospf monitor lsdb destination** *<hostname-or-IPaddr>*
Shows information about all OSPF routing neighbors.	**ospf monitor neighborsdestination** *<hostname-or-IPaddr>*
Show information on valid next hops.	**ospf monitor next-hop-list destination** *<hostname-or-IPaddr>*
Display OSPF routing table.	**ospf monitor routes destination** *<hostname-or-IPaddr>*
Monitor OSPF statistics for a specified destination.	**ospf monitor statistics destination** *<hostname-or-IPaddr>*
Shows information about all OSPF routing version	**ospf monitor version**
Shows OSPF Autonomous System External Link State Database.	**ospf sbow AS-External-LSDB**
Show all OSPF tables.	**ospf show all**
Show all OSPF areas.	**ospf show areas**
Show OSPF errors.	**ospf show errors**
Show information about OSPF export policies.	**ospf show export-policies**

Shows routes redistributed into OSPF.	**ospf show exported-routes**
Show all OSPF global parameters.	**ospf show globals**
Show information about OSPF import policies.	**ospf show import-policies**
Show OSPF interfaces.	**ospf show interfaces**
Shows information about all valid next hops mostly derived from the SPF calculation.	**ospf show next-hop-list**
Show OSPF statistics.	**ospf show statistics**
Shows information about OSPF Border Routes.	**ospf show summary-asb**
Show OSPF timers.	**ospf show timers**
Show OSPF virtual-links.	**ospf show virtual-links**

OSPF Configuration Examples

For all examples in this section, refer to the configuration shown in Figure 1 on page 5 - 12.

The following configuration commands for router R1:

- Determine the IP address for each interface
- Specify the static routes configured on the router
- Determine its OSPF configuration

```
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Create the various IP interfaces.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  140.1.3.1/24 port et.1.6
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Configure default routes to the other subnets reachable through R2.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.1.1.2
   ip add route 160.1.5.0/24 gateway 120.1.1.2
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! OSPF Box Level Configuration
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf start
   ospf create area 140.1.0.0
   ospf create area backbone
   ospf set ase-defaults cost 4
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! OSPF Interface Configuration
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf add interface 140.1.1.1 to-area 140.1.0.0
   ospf add interface 140.1.2.1 to-area 140.1.0.0
   ospf add interface 140.1.3.1 to-area 140.1.0.0
   ospf add interface 130.1.1.1 to-area backbone
```

Exporting all interface and static routes to OSPF

Router R1 has several static routes. We would export these static routes as type-2 OSPF routes. The interface routes would be redistributed as type-1 OSPF routes.

1. Create a OSPF export destination for type-1 routes since we would like to redistribute certain routes into OSPF as type 1 OSPF-ASE routes.

   ```
   ip-router policy create ospf-export-destination
      ospfExpDstType1 type 1 metric 1
   ```

2. Create a OSPF export destination for type-2 routes since we would like to redistribute certain routes into OSPF as type 2 OSPF-ASE routes.

   ```
   ip-router policy create ospf-export-destination
      ospfExpDstType2 type 2 metric 4
   ```

3. Create a Static export source since we would like to export static routes.

   ```
   ip-router policy create static-export-source statExpSrc
   ```

4. Create a Direct export source since we would like to export interface/direct routes.

   ```
   ip-router policy create direct-export-source directExpSrc
   ```

5. Create the Export-Policy for redistributing all interface routes and static routes into OSPF.

   ```
   ip-router policy export destination ospfExpDstType1 source
      directExpSrc network all
   ```

   ```
   ip-router policy export destination ospfExpDstType2 source
      statExpSrc network all
   ```

Exporting all RIP, interface, and static routes to OSPF

Note: Also export interface, static, RIP, OSPF, and OSPF-ASE routes into RIP.

In the configuration shown in Figure 1 on page 5 - 12, suppose if we decide to run RIP Version 2 on network 120.190.0.0/16, connecting routers R1 and R2.

We would like to redistribute these RIP routes as OSPF type-2 routes, and associate the tag 100 with them. Router R1 would also like to redistribute its static routes as type 2 OSPF routes. The interface routes would redistributed as type 1 OSPF routes.

Router R1 would like to redistribute its OSPF, OSPF-ASE, RIP, Static and Interface/ Direct routes into RIP.

1. Enable RIP on interface 120.190.1.1/16.

   ```
   rip add interface 120.190.1.1
   ```

   ```
   rip set interface 120.190.1.1 version 2 type multicast
   ```

2. Create a OSPF export destination for type-1 routes.

   ```
   ip-router policy create ospf-export-destination
      ospfExpDstType1 type 1 metric 1
   ```

3. Create a OSPF export destination for type-2 routes.

   ```
   ip-router policy create ospf-export-destination
      ospfExpDstType2 type 2 metric 4
   ```

4. Create a OSPF export destination for type-2 routes with a tag of 100.

   ```
   ip-router policy create ospf-export-destination
      ospfExpDstType2t100 type 2 tag 100 metric 4
   ```

5. Create a RIP export source.

   ```
   ip-router policy export destination ripExpDst source
      ripExpSrc network all
   ```

6. Create a Static export source.

   ```
   ip-router policy create static-export-source statExpSrc
   ```

7. Create a Direct export source.

   ```
   ip-router policy create direct-export-source directExpSrc
   ```

8. Create the Export-Policy for redistributing all interface, RIP and static routes into OSPF.

   ```
   ip-router policy export destination ospfExpDstType1 source
      directExpSrc network all
   ```

   ```
   ip-router policy export destination ospfExpDstType2 source
      statExpSrc network all
   ```

   ```
   ip-router policy export destination ospfExpDstType2t100
      source ripExpSrc network all
   ```

9. Create a RIP export destination.

```
ip-router policy create rip-export-destination ripExpDst
```

10. Create OSPF export source.

```
ip-router policy create ospf-export-source ospfExpSrc type OSPF
```

11. Create OSPF-ASE export source.

```
ip-router policy create ospf-export-source ospfAseExpSrc type
   OSPF-ASE
```

12. Create the Export-Policy for redistributing all interface, RIP, static, OSPF and OSPF-ASE routes into RIP.

```
ip-router policy export destination ripExpDst source
   statExpSrc network all
```

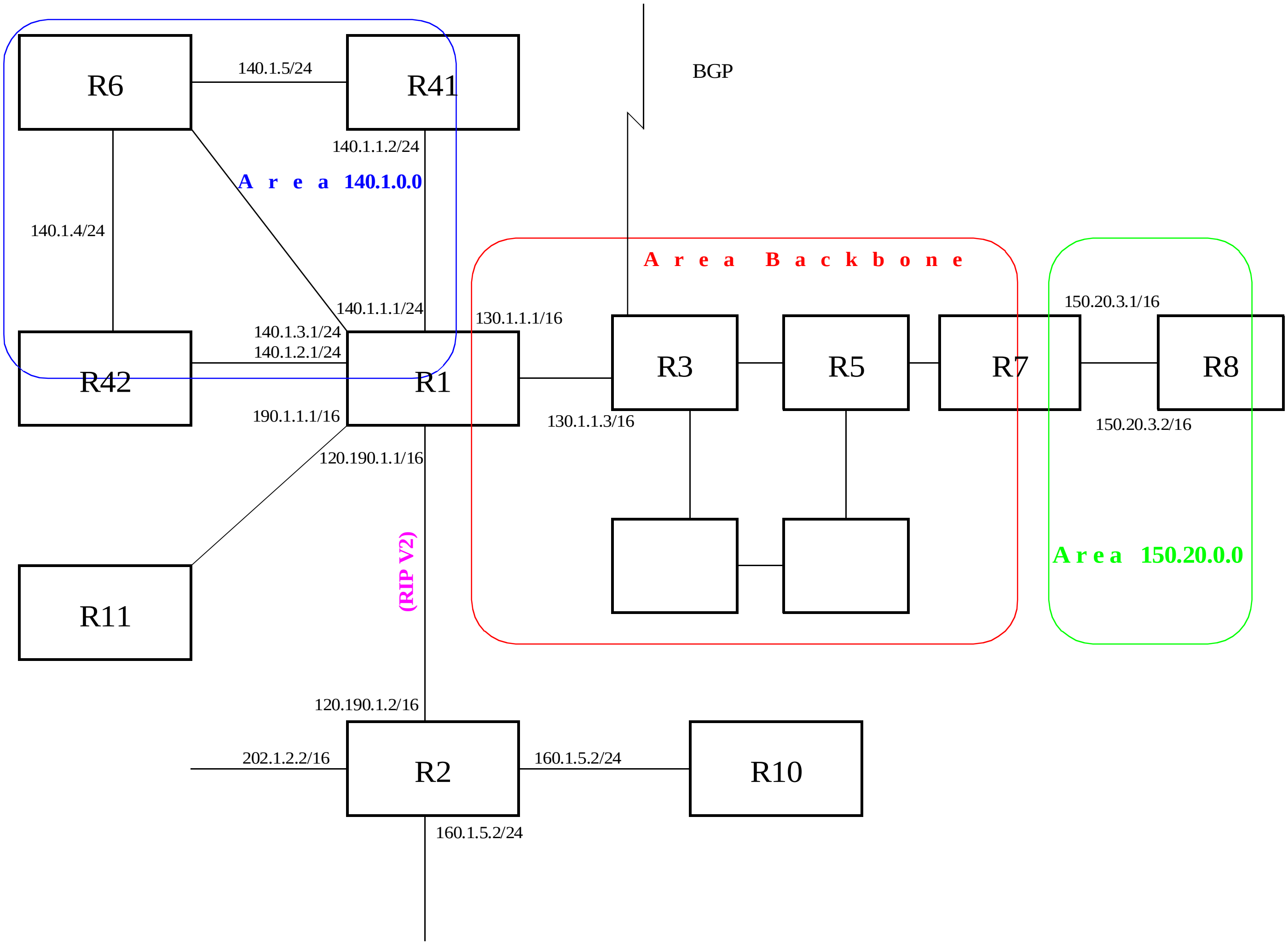
```
ip-router policy export destination ripExpDst source
   ripExpSrc network all
```

```
ip-router policy export destination ripExpDst source
   directExpSrc network all
```

```
ip-router policy export destination ripExpDst source
   ospfExpSrc network all
```

```
ip-router policy export destination ripExpDst source
   ospfAseExpSrc network all
```

Figure 1: Exporting to OSPF



Chapter 6   Routing Policy Configuration Guide

Route Import and Export Policy Overview

The GIGAswitch/Router supports extremely flexible routing policies. The GIGAswitch/Router allows the network administrator to control import and export of routing information based on criteria including:

- Individual protocol
- Source and destination autonomous system
- Source and destination interface
- Previous hop router
- Autonomous system path
- Tag associated with routes
- Specific destination address

The network administrator can specify a preference level for each combination of routing information being imported by using a flexible masking capability.

The GIGAswitch/Router also provides the ability to create advanced and simple routing policies. Simple routing policies provide a quick route redistribution between various routing protocols (RIP and OSPF). Advanced routing policies provide more control over route redistribution.

Preference

Preference is the value the GIGAswitch/Router routing process uses to order preference of routes from one protocol or peer over another. Preference can be set using several different configuration commands. Preference can be set based on one network interface over another, from one protocol over another, or from one remote gateway over another. Preference may not be used to control the selection of routes within an Interior Gateway Protocol (IGP) This is accomplished automatically by the protocol based on metric.

Preference may be used to select routes from the same Exterior Gateway Protocol (EGP) learned from different peers or autonomous systems. Each route has only one preference value associated with it, even though the preference can be set at many places using configuration commands. The last or most specific preference value set for a route is the value used. A preference value is an arbitrarily assigned value used to determine the order of routes to the same destination in a single routing database. The active route is chosen by the lowest preference value.

A default preference is assigned to each source from which the GIGAswitch/Router routing process receives routes. Preference values range from 0 to 255 with the lowest number indicating the most preferred route.

The following table summarizes the default preference values for routes learned in various ways. The table lists the CLI commands that set preference, and shows the types of routes to which each CLI command applies. A default preference for each type of route is listed, and the table notes preference precedence between protocols. The narrower the scope of the statement, the higher precedence its preference value is given, but the smaller the set of routes it affects.

Preference of	Defined by CLI Command	Default
Direct connected networks	ip-router global set interface	0
OSPF routes	ospf	10
Static routes from config	ip add route	60
RIP routes	rip set preference	100
Point-to-point interface		110
Routes to interfaces that are down	ip-router global set interface down-preference	120
Aggregate/generate routes	aggr-gen	130
OSPF AS external routes	ospf set ase-defaults preference	150
BGP routes	bgp set preference	170

Import Policies

Import policies control the importation of routes from routing protocols and their installation in the routing databases (Routing Information Base and Forwarding Information Base). Import Policies determine which routes received from other systems are used by the GIGAswitch/Router routing process. Every import policy can have up to two components:

- Import-Source
- Route-Filter

Import-Source

This component specifies the source of the imported routes. It can also specify the preference to be associated with the routes imported from this source.

The routes to be imported can be identified by their associated attributes:

- Type of the source protocol (RIP, OSPF, BGP).
- Source interface or gateway from which the route was received.
- Source autonomous system from which the route was learned.
- AS path associated with a route. Besides autonomous system, BGP also supports importation of routes using AS path regular expressions, and AS path options.

  If multiple communities are specified using the optional-attributes-list, only updates carrying all of the specified communities will be matched. If the specified optional-attributes-list has the value **none** for the **well-known-community** option, then only updates lacking the community attribute will be matched.

In some cases, a combination of the associated attributes can be specified to identify the routes to be imported.

Note: It is quite possible for several BGP import policies to match a given update. If more than one policy matches, the first matching policy will be used. All later matching policies will be ignored. For this reason, it is generally desirable to order import policies from most to least specific. An import policy with an optional-attributes-list will match any update with any (or no) communities.

The importation of RIP routes may be controlled by source interface and source gateway. RIP does not support the use of preference to choose between RIP routes. That is left to the protocol metrics.

Due to the nature of OSPF, only the importation of ASE routes may be controlled. OSPF intra-and inter-area routes are always imported into the routing table with a preference of 10. If a tag is specified with the import policy, routes with the specified tag will only be imported.

It is only possible to restrict the importation of OSPF ASE routes when functioning as an AS border router.

Like the other interior protocols, preference cannot be used to choose between OSPF ASE routes. That is done by the OSPF costs.

Route-Filter

This component specifies the individual routes which are to be imported or restricted. The preference to be associated with these routes can also be explicitly specified using this component.

The preference associated with the imported routes are inherited unless explicitly specified. If there is no preference specified with a route-filter, then the preference is inherited from the one specified with the import-source.

Every protocol (RIP, OSPF, and BGP) has a configurable parameter that specifies default-preference associated with routes imported to that protocol. If a preference is not explicitly specified with the route-filter, as well as the import-source, then it is inherited from the default-preference associated with the protocol for which the routes are being imported.

Export Policies

Export policies control the redistribution of routes to other systems. They determine which routes are advertised by the Unicast Routing Process to other systems. Every export policy can have up to three components:

- Export-Destination
- Export-Source
- Route-Filter

Export-Destination

This component specifies the destination where the routes are to be exported. It also specifies the attributes associated with the exported routes. The interface, gateway or the autonomous system to which the routes are to be redistributed are a few examples of export-destinations. The metric, type, tag, and AS-Path are a few examples of attributes associated with the exported routes.

Export-Source

This component specifies the source of the exported routes. It can also specify the metric to be associated with the routes exported from this source.

The routes to be exported can be identified by their associated attributes:

- Their protocol type (RIP, OSPF, BGP, Static, Direct, Aggregate).
- Interface or the gateway from which the route was received.
- Autonomous system from which the route was learned.
- AS path associated with a route. When BGP is configured, all routes are assigned an AS path when they are added to the routing table. For interior routes, this AS path

specifies IGP as the origin and no ASs in the AS path (the current AS is added when the route is exported). For BGP routes, the AS path is stored as learned from BGP.

- Tag associated with a route. Both OSPF and RIP version 2 currently support tags. All other protocols have a tag of zero.

In some cases, a combination of the associated attributes can be specified to identify the routes to be exported.

Route-Filter

This component specifies the individual routes which are to exported or restricted. The metric to be associated with these routes can also be explicitly specified using this component.

The metric associated with the exported routes are inherited unless explicitly specified. If there is no metric specified with a route-filter, then the metric is inherited from the one specified with the export-source.

If a metric was not explicitly specified with both the route-filter and the export-source, then it is inherited from the one specified with the export-destination.

Every protocol (RIP, OSPF, and BGP) has a configurable parameter that specifies default-metric associated with routes exported to that protocol. If a metric is not explicitly specified with the route-filter, export-source as well as export-destination, then it is inherited from the default-metric associated with the protocol to which the routes are being exported.

Specifying a Route Filter

Routes are filtered by specifying a route-filter that will match a certain set of routes by destination, or by destination and mask. Among other places, route filters are used with martians and in import and export policies.

The action taken when no match is found is dependent on the context. For instance, a route that does match any of the route-filters associated with the specified import or export policies is rejected.

A route will match the most specific filter that applies. Specifying more than one filter with the same destination, mask and modifiers generates an error.

There are three possible formats for a route filter. Not all of these formats are available in all places. In most cases, it is possible to associate additional options with a filter. For example, while creating a martian, it is possible to specify the **allow** option, while creating an import policy, one can specify a **preference**, and while creating an export policy one can specify a **metric**.

The three forms of a route-filter are:

- Network [exact | refines | between number,number]
- Network/mask [exact | refines | between number,number]
- Network/masklen [exact | refines | between number,number]

Matching usually requires both an address and a mask, although the mask is implied in the shorthand forms listed below. These three forms vary in how the mask is specified. In the first form, the mask is implied to be the natural mask of the network. In the second, the mask is explicitly specified. In the third, the mask is specified by the number of contiguous one bits.

If no optional parameters (exact, refines, or between) are specified, any destination that falls in the range given by the network and mask is matched, so the mask of the destination is ignored. If a natural network is specified, the network, any subnets, and any hosts will be matched. Three optional parameters that cause the mask of the destination to also be considered are:

- **Exact:** Specifies that the mask of the destination must match the supplied mask exactly. This is used to match a network, but no subnets or hosts of that network.
- **Refines:** Specifies that the mask of the destination must be more specified (i.e., longer) than the filter mask. This is used to match subnets and/or hosts of a network, but not the network.
- **Between number, number:** Specifies that the mask of the destination must be as or more specific (i.e., as long as or longer) than the lower limit (the first number parameter) and no more specific (i.e., as long as or shorter) than the upper limit (the second number). Note that exact and refines are both special cases of between.

Aggregates and Generates

Route aggregation is a method of generating a more general route, given the presence of a specific route. It is used, for example, at an autonomous system border to generate a route to a network to be advertised via BGP given the presence of one or more subnets of that network learned via OSPF. The routing process does not perform any aggregation unless explicitly requested.

Route aggregation is also used by regional and national networks to reduce the amount of routing information passed around. With careful allocation of network addresses to clients, regional networks can just announce one route to regional networks instead of hundreds.

Aggregate routes are not actually used for packet forwarding by the originator of the aggregate route, but only by the receiver (if it wishes). Instead of requiring a route-peer to know about individual subnets which would increase the size of its routing table, the peer is only informed about an aggregate-route which contains all the subnets.

Like export policies, aggregate-routes can have up to three components:

- Aggregate-Destination
- Aggregate-Source
- Route-Filter

Aggregate-Destination

This component specifies the aggregate/summarized route. It also specifies the attributes associated with the aggregate route. The preference to be associated with an aggregate route can be specified using this component.

Aggregate-Source

This component specifies the source of the routes contributing to an aggregate/ summarized route. It can also specify the preference to be associated with the contributing routes from this source. This preference can be overridden by explicitly specifying a preference with the route-filter.

The routes contributing to an aggregate can be identified by their associated attributes:

- Protocol type (RIP, OSPF, BGP, Static, Direct, Aggregate).
- Autonomous system from which the route was learned.
- AS path associated with a route. When BGP is configured, all routes are assigned an AS path when they are added to the routing table. For interior routes, this AS path specifies IGP as the origin and no ASs in the AS path (the current AS is added when the route is exported). For BGP routes, the AS path is stored as learned from BGP.
- Tag associated with a route. Both OSPF and RIP version 2 currently support tags. All other protocols have a tag of zero.

In some cases, a combination of the associated attributes can be specified to identify the routes contributing to an aggregate.

Route-Filter

This component specifies the individual routes that are to be aggregated or summarized. The preference to be associated with these routes can also be explicitly specified using this component.

The contributing routes are ordered according to the aggregation preference that applies to them. If there is more than one contributing route with the same aggregating preference, the route's own preferences are used to order the routes. The preference of the aggregate route will be that of contributing route with the lowest aggregate preference.

A route may only contribute to an aggregate route that is more general than itself; it must match the aggregate under its mask. Any given route may only contribute to one aggregate route, which will be the most specific configured, but an aggregate route may contribute to a more general aggregate.

An aggregate-route only comes into existence if at least one of its contributing routes is active.

Authentication

Authentication guarantees that routing information is only imported from trusted routers. Many protocols like RIP V2 and OSPF provide mechanisms for authenticating protocol exchanges. A variety of authentication schemes can be used. Authentication has two components – an Authentication Method and an Authentication Key. Many protocols allow different authentication methods and keys to be used in different parts of the network.

Authentication Methods

There are mainly two authentication methods:

Simple Password: In this method, an authentication key of up to 8 characters is included in the packet. If this does not match what is expected, the packet is discarded. This method provides little security, as it is possible to learn the authentication key by watching the protocol packets.

MD5: This method uses the MD5 algorithm to create a crypto-checksum of the protocol packet and an authentication key of up to 16 characters. The transmitted packet does not contain the authentication key itself, instead it contains a crypto-checksum, called the digest. The receiving router performs a calculation using the correct authentication key and discard the packet if the digest does not match. In addition, a sequence number is maintained to prevent the replay of older packets. This method provides a much stronger assurance that routing data originated from a router with a valid authentication key.

Many protocols allow the specification of two authentication keys per interface. Packets are always sent using the primary keys, but received packets are checked with both the primary and secondary keys before being discarded.

Authentication Keys and Key Management

An authentication key permits generation and verification of the authentication field in protocol packets. In many situations, the same primary and secondary keys are used on several interfaces of a router. For ease of management of keys, a concept of key-chain is introduced. Each key-chain has an identifier and contains up to two keys. One of keys is the primary key and other is the secondary key. Outgoing packets use the primary authentication key, but incoming packets may match either the primary or secondary authentication key. In the router configuration mode, instead of specifying the key for each interface (which can be up to 16 characters long), a key-chain identifier is specified.

Configuring Simple Routing Policies

Simple routing policies provide an efficient way for routing information to be exchanged between routing protocols. The **redistribute** command can be used to redistribute routes from one routing domain into another routing domain. Redistribution of routes between routing domains is based on route policies. A route policy is a set of conditions based on which routes are redistributed. While the redistribute command is expected to satisfy the export policy requirement for most users, complex export policies may require the use of the commands listed under Export Policies.

The general syntax of the redistribute command is as follows:

```
ip-router policy redistribute from-proto <protocol> to-proto <protocol>
    [network <ipAddr-mask> [exact|refines|between <low-high>]] [metric
    <number>|restrict] [source-as <number>] [target-as <number>]
```

The from-proto parameter specifies the protocol of the source routes. The values for the from-proto parameter are rip, ospf, bgp, direct, static, aggregate and ospf-ase. The to-proto parameter specifies the destination protocol where the routes are to be exported. The values for the to-proto parameter are rip, ospf and bgp. The network parameter provides a means to define a filter for the routes to be distributed. The network parameter defines a filter that is made up of an IP address and a mask. Routes that match the filter are considered as eligible for redistribution.

Every protocol (RIP, OSPF, and BGP) has a configurable parameter that specifies default-metric associated with routes exported to that protocol. If a metric is not explicitly specified with the redistribute command, then it is inherited from the default-metric associated with the protocol to which the routes are being exported.

Redistributing Static Routes

Static routes may be redistributed to another routing protocol such as RIP or OSPF by the following command. The **network** parameter specifies the set of static routes that will be redistributed by this command. If all static routes are to be redistributed set the **network** parameter to **all**. Note that the **network** parameter is a filter that is used to specify routes that are to be redistributed.

To redistribute static routes, enter one of the following commands in Configure mode:

To redistribute static routes into RIP.	**ip-router policy redistribute from-proto static to-proto rip network all**
To redistribute static routes into OSPF.	**ip-router policy redistribute from-proto static to-proto ospf network all**

Redistributing Directly Attached Networks

Routes to directly attached networks are redistributed to another routing protocol such as RIP or OSPF by the following command. The **network** parameter specifies a set of routes that will be redistributed by this command. If all direct routes are to be redistributed set the **network** parameter to **all**. Note that the **network** parameter is a filter that is used to specify routes that are to be redistributed.

To redistribute direct routes, enter one of the following commands in Configure mode:

To redistribute direct routes into RIP.	**ip-router policy redistribute from-proto direct to-proto rip network all**
To redistribute direct routes into OSPF.	**ip-router policy redistribute from-proto direct to-proto ospf network all**

Redistributing RIP into RIP

The GIGAswitch/Router routing process requires RIP redistribution into RIP if a protocol is redistributed into RIP.

To redistribute RIP into RIP, enter the following command in Configure mode:

To redistribute RIP into RIP.	**ip-router policy redistribute from-proto rip to-proto rip**

Redistributing RIP into OSPF

RIP routes may be redistributed to OSPF.

To redistribute RIP into OSPF, enter the following command in Configure mode:

To redistribute RIP into OSPF.	**ip-router policy redistribute from-proto rip to-proto ospf**

Redistributing OSPF to RIP

For the purposes of route redistribution and import-export policies, OSPF intra- and inter-area routes are referred to as **ospf** routes, and external routes redistributed into OSPF are referred to as **ospf-ase** routes. Examples of **ospf-ase** routes include **static** routes, **rip** routes, **direct** routes, **bgp** routes, or **aggregate** routes, which are redistributed into an OSPF domain.

OSPF routes may be redistributed into RIP. To redistribute OSPF into RIP, enter the following command in Configure mode:

To redistribute **ospf-ase** routes into **rip**.	**`ip-router policy redistribute from-proto ospf-ase to-proto rip`**
To redistribute **ospf** routes into **rip**.	**`ip-router policy redistribute from-proto ospf to-proto rip`**

Redistributing Aggregate Routes

The **aggregate** parameter causes an aggregate route with the specified IP address and subnet mask to be redistributed.

Note: The aggregate route must first be created using the **aggr-gen** command**.** This command creates a specified aggregate route for routes that match the aggregate.

To redistribute aggregate routes, enter one of the following commands in Configure mode:

To redistribute aggregate routes into RIP.	**`ip-router policy redistribute from-proto aggregate to-proto rip`**
To redistribute aggregate routes into OSPF.	**`ip-router policy redistribute from-proto aggregate to-proto OSPF`**

Simple Route Redistribution Examples

Example 1: Redistribution into RIP

For all examples given in this section, refer to the configurations shown in Figure 2 on page 6 - 21.

The following configuration commands for router R1:

- Determine the IP address for each interface

- Specify the static routes configured on the router
- Determine its RIP configuration

```
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Create the various IP interfaces.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  160.1.1.1/16 port et.1.6
   interface create ip to-r7  address-netmask  170.1.1.1/16 port et.1.7
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure a default route through 170.1.1.7
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route default gateway 170.1.1.7
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure static routes to the 135.3.0.0 subnets reachable through
! R3.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 135.3.1.0/24 gateway 130.1.1.3
   ip add route 135.3.2.0/24 gateway 130.1.1.3
   ip add route 135.3.3.0/24 gateway 130.1.1.3
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  ! Configure default routes to the other subnets reachable through R2.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.190.1.2
   ip add route 160.1.5.0/24 gateway 120.190.1.2
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! RIP Box Level Configuration
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   rip start
   rip set default-metric 2
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! RIP Interface Configuration. Create a RIP interfaces, and set
   ! their type to (version II, multicast).
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   rip add interface to-r41
   rip add interface to-r42
   rip add interface to-r6
   rip set interface to-r41 version 2 type multicast
   rip set interface to-r42 version 2 type multicast
   rip set interface to-r6  version 2 type multicast
```

Exporting a given static route to all RIP interfaces

Router R1 has several static routes of which one is the default route. We would export this default route over all RIP interfaces.

```
ip-router policy redistribute from-proto static to-proto rip network default
```

Exporting all static routes to all RIP interfaces

Router R1 has several static routes. We would export these routes over all RIP interfaces.

```
ip-router policy redistribute from-proto static to-proto rip network all
```

Exporting all static routes except the default route to all RIP interfaces

Router R1 has several static routes. We would export all these routes except the default route to all RIP interfaces.

```
ip-router policy redistribute from-proto static to-proto rip network all
```

```
ip-router policy redistribute from-proto static to-proto rip network default
     restrict
```

Example 2: Redistribution into OSPF

For all examples given in this section, refer to the configurations shown in Figure 3 on page 6 - 25.

The following configuration commands for router R1:

- Determine the IP address for each interface
- Specify the static routes configured on the router
- Determine its OSPF configuration

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Create the various IP interfaces.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  140.1.3.1/24 port et.1.6
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Configure default routes to the other subnets reachable through R2.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.1.1.2
   ip add route 160.1.5.0/24 gateway 120.1.1.2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! OSPF Box Level Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf start
```

```
      ospf create area 140.1.0.0
      ospf create area backbone
      ospf set ase-defaults cost 4
   !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      ! OSPF Interface Configuration
   !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      ospf add interface 140.1.1.1 to-area 140.1.0.0
      ospf add interface 140.1.2.1 to-area 140.1.0.0
      ospf add interface 140.1.3.1 to-area 140.1.0.0
      ospf add interface 130.1.1.1 to-area backbone
```

Exporting all interface and static routes to OSPF.

Router R1 has several static routes. We would like to export all these static routes and direct-routes (routes to connected networks) into OSPF.

```
ip-router policy redistribute from-proto static to-proto ospf

ip-router policy redistribute from-proto direct to-proto ospf
```

Note: The network parameter specifying the network-filter is optional. The default value for this parameter is **all**, indicating all networks. Since in the above example, we would like to export all static and direct routes into OSPF, we have not specified this parameter.

Exporting all RIP, interface, and static routes to OSPF

Note: Also export interface, static, RIP, OSPF, and OSPF-ASE routes into RIP.

In the configuration shown in Figure 3 on page 6 - 25, suppose if we decide to run RIP Version 2 on network 120.190.0.0/16, connecting routers R1 and R2.

Router R1 would like to export all RIP, interface, and static routes to OSPF.

```
ip-router policy redistribute from-proto rip to-proto ospf

ip-router policy redistribute from-proto direct to-proto ospf

ip-router policy redistribute from-proto static to-proto ospf
```

Router R1 would also like to export interface, static, RIP, OSPF, and OSPF-ASE routes into RIP.

```
ip-router policy redistribute from-proto direct to-proto rip

ip-router policy redistribute from-proto static to-proto rip

ip-router policy redistribute from-proto rip to-proto rip

ip-router policy redistribute from-proto ospf to-proto rip

ip-router policy redistribute from-proto ospf-ase to-proto rip
```

Configuring Advanced Routing Policies

Advanced Routing Policies are used for creating complex import/export policies that cannot be done using the redistribute command. Advanced export policies provide granular control over the targets where the routes are exported, the source of the exported routes, and the individual routes which are exported. It provides the capability to send different routes to the various route-peers. They can be used to provide the same route with different attributes to the various route-peers.

Import policies control the importation of routes from routing protocols and their installation in the routing database (Routing Information Base and Forwarding Information Base). Import policies determine which routes received from other systems are used by the GIGAswitch/Router routing process. Using import policies, it is possible to ignore route updates from an unreliable peer and give better preference to routes learned from a trusted peer.

Export Policies

Advanced export policies can be constructed from one or more of the following building blocks:

- Export Destinations - This component specifies the destination where the routes are to be exported. It also specifies the attributes associated with the exported routes. The interface, gateway or the autonomous system to which the routes are to be redistributed are a few examples of export-destinations. The metric, type, tag, and AS-Path are a few examples of attributes associated with the exported routes.
- Export Sources - This component specifies the source of the exported routes. It can also specify the metric to be associated with the routes exported from this source. The routes to be exported can be identified by their associated attributes, such as protocol type, interface or the gateway from which the route was received, and so on.
- Route Filter - This component provides the means to define a filter for the routes to be distributed. Routes that match a filter are considered as eligible for redistribution. This can be done using one of two methods:
    - Creating a route-filter and associating an identifier with it. A route-filter has several network specifications associated with it. Every route is checked against the set of network specifications associated with all route-filters to determine its eligibility for redistribution. The identifier associated with a route-filter is used in the *ip-router policy export* command.
    - Specifying the networks as needed in the *ip-router policy export* command.

  If you want to create a complex route-filter, and you intend to use that route-filter in several export policies, then the first method is recommended. It you do not have complex filter requirements, then use the second method.

After you create one or more building blocks, they are tied together by the *iprouter policy export* command.

To create route export policies, enter the following command in Configure mode:

Create an export policy.	**ip-router policy export destination** *<exp-dest-id>* **[source** *<exp-src-id>* **[filter** *<filter-id>***|[network** *<ipAddr-mask>* **[exact|refines|between** *<low-high>***]** **[metric** *<number>***|restrict]]]]**

The *<exp-dest-id>* is the identifier of the export-destination which determines where the routes are to be exported. If no routes to a particular destination are to be exported, then no additional parameters are required.

The *<exp-src-id>*, if specified, is the identifier of the export-source which determines the source of the exported routes. If a export-policy for a given export-destination has more than one export-source, then the *ip-router policy export destination <exp-dest-id>* command should be repeated for each *<exp-src-id>*.

The *<filter-id>*, if specified, is the identifer of the route-filter associated with this export-policy. If there is more than one route-filter for any export-destination and export-source combination, then the *ip-router policy export destination <exp-dest-id> source <exp-src-id>* command should be repeated for each *<filter-id>*.

Creating an Export Destination

To create an export destination, enter one the following commands in Configure mode:

Create a RIP export destination.	**ip-router policy create rip-export-destination** *<name>*
Create an OSPF export destination.	**ip-router policy create ospf-export-destination** *<name>*

Creating an Export Source

To create an export source, enter one of the following commands in Configure mode:

Create a RIP export source.	**ip-router policy create rip-export-source** *<name>*
Create an OSPF export source.	**ip-router policy create ospf-export-source** *<name>*

Import Policies

Import policies can be constructed from one or more of the following building blocks:

- Import-source - This component specifies the source of the imported routes. It can also specify the preference to be associated with the routes imported from this source. The routes to be imported can be identified by their associated attributes, including source protocol, Source interface or gateway from which the route was received, and so on.
- Route Filter - This component provides the means to define a filter for the routes to be imported. Routes that match a filter are considered as eligible for importation. This can be done using one of two methods:
  - Creating a route-filter and associating an identifier with it. A route-filter has several network specifications associated with it. Every route is checked against the set of network specifications associated with all route-filters to determine its eligibility for importation. The identifier associated with a route-filter is used in the *ip-router policy import* command.
  - Specifying the networks as needed in the *ip-router policy import* command.

  If you want to create a complex route-filter, and you intend to use that route-filter in several import policies, then the first method is recommended. It you do not have complex filter requirements, then use the second method.

After you create one or more building blocks, they are tied together by the *iprouter policy import* command.

To create route import policies, enter the following command in Configure mode:

Create an import policy.	**ip-router policy import source** *<imp-src-id>* **[filter** *<filter-id>***|[network** *<ipAddr-mask>* **[exact|refines|between** *<low-high>***] [preference** *<number>***|restrict]]]**

The *<imp-src-id>* is the identifier of the import-source that determines the source of the imported routes. If no routes from a particular source are to be imported, then no additional parameters are required.

The *<filter-id>*, if specified, is the identifer of the route-filter associated with this import-policy. If there is more than one route-filter for any import-source, then the *ip-router policy import source <imp-src-id>* command should be repeated for each *<filter-id>*.

Creating an Import Source

Import sources specify the routing protocol from which the routes are imported. The source may be RIP or OSPF.

To create an import source, enter one of the following commands in Configure mode:

Create a RIP import destination.	**ip-router policy create rip-import-source** *<name>*
Create an OSPF import destination.	**ip-router policy create ospf-import-source** *<name>*

Creating a Route Filter

Route policies are defined by specifying a set of filters that will match a certain route by destination, or by destination and mask.

To create route filters, enter the following command in Configure mode:

Create a route filter.	**ip-router policy create filter** *<name-id>* **network** *<IP-address/mask>*

Creating an Aggregate Route

Route aggregation is a method of generating a more general route, given the presence of a specific route. The routing process does not perform any aggregation unless explicitly requested. Aggregate-routes can be constructed from one or more of the following building blocks:

- Aggregate-Destination - This component specifies the aggregate/summarized route. It also specifies the attributes associated with the aggregate route. The preference to be associated with an aggregate route can be specified using this component.
- Aggregate-Source - This component specifies the source of the routes contributing to an aggregate/summarized route. It can also specify the preference to be associated

with the contributing routes from this source. The routes contributing to an aggregate can be identified by their associated attributes, including protocol type, tag associated with a route, and so on.

- Route Filter - This component provides the means to define a filter for the routes to be aggregated or summarized. Routes that match a filter are considered as eligible for aggregation. This can be done using one of two methods:
  - Creating a route-filter and associating an identifier with it. A route-filter has several network specifications associated with it. Every route is checked against the set of network specifications associated with all route-filters to determine its eligibility for aggregation. The identifier associated with a route-filter is used in the *ip-router policy aggr-gen* command.
  - Specifying the networks as needed in the *ip-router policy aggr-gen* command.

  If you want to create a complex route-filter, and you intend to use that route-filter in several aggregates, then the first method is recommended. It you do not have complex filter requirements, then use the second method.

After you create one or more building blocks, they are tied together by the *iprouter policy aggr-gen* command.

To create aggregates, enter the following command in Configure mode:

Create an aggregate route.	**ip-router policy aggr-gen destination** *<aggr-dest-id>* **[source** *<aggr-src-id>* **[filter** *<filter-id>***|[network** *<ipAddr-mask>* **[exact|refines|between** *<low-high>***] [preference** *<number>***|restrict]]]]**

The *<aggr-dest-id>* is the identifier of the aggregate-destination that specifies the aggregate/summarized route.

The *<aggr-src-id>* is the identifier of the aggregate-source that contributes to an aggregate route. If an aggregate has more than one aggregate-source, then the *ip-router policy aggr-gen destination <aggr-dest-id>* command should be repeated for each *<aggr-src-id>*.

The *<filter-id>* is the identifer of the route-filter associated with this aggregate. If there is more than one route-filter for any aggregate-destination and aggregate-source combination, then the *ip-router policy aggr-gen destination <aggr-dest-id> source <aggr-src-id>* command should be repeated for each *<filter-id>*.

Creating an Aggregate Destination

To create an aggregate destination, enter the following command in Configure mode:

Create an aggregate destination.	**`ip-router policy create aggr-gen-dest`** *`<name>`* **`network`** *`<ipAddr-mask>`*

Creating an Aggregate Source

To create an aggregate source, enter the following command in Configure mode:

Create an aggregate source.	**`ip-router policy create aggr-gen-source`** *`<name>`* **`protocol`** *`<protocol-name>`*

Examples of Import Policies

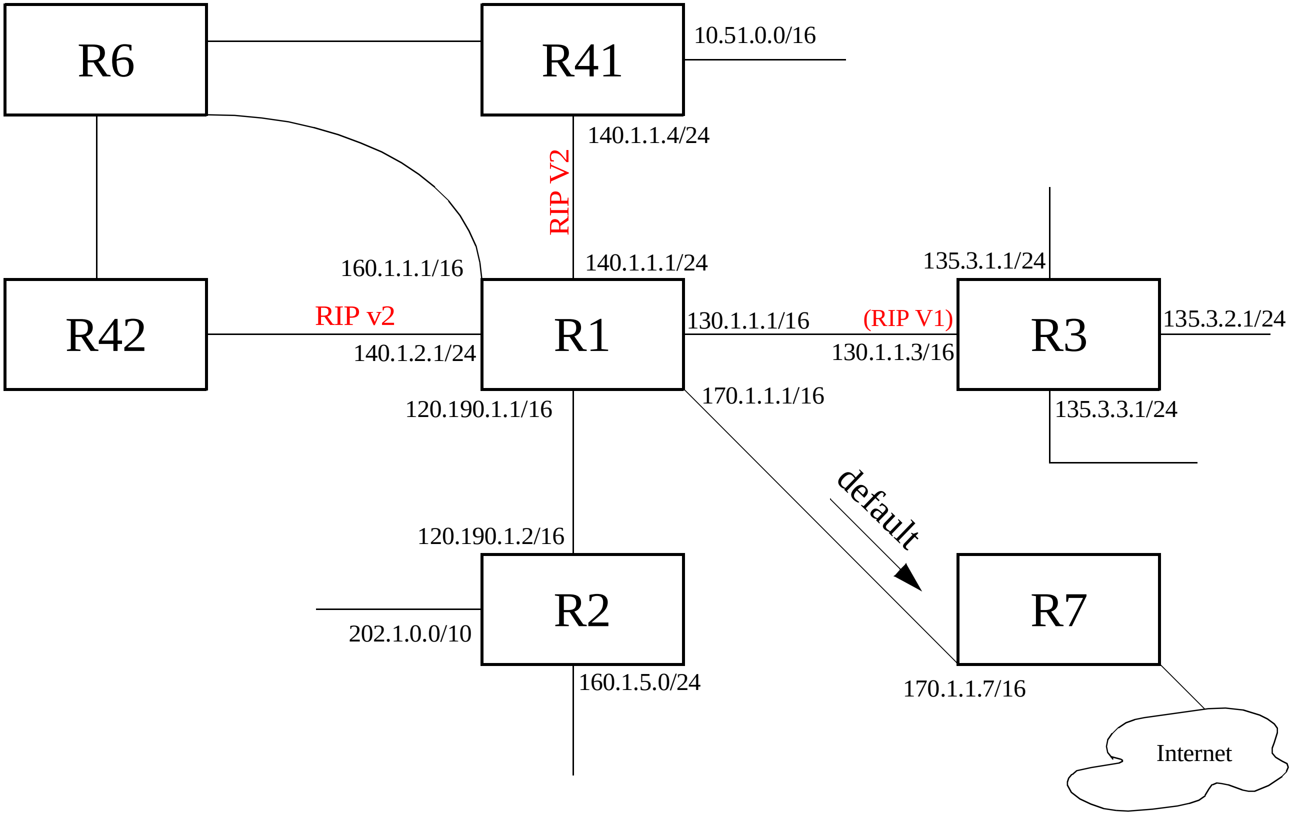
Example 1: Importing from RIP

The importation of RIP routes may be controlled by any of protocol, source interface, or source gateway. If more than one is specified, they are processed from most general (protocol) to most specific (gateway).

RIP does not support the use of preference to choose between routes of the same protocol. That is left to the protocol metrics.

For all examples in this section, refer to the configuration shown in Figure 2 on page 6 - 21.

Figure 2: Exporting to RIP



The following configuration commands for router R1

- Determine the IP address for each interface.
- Specify the static routes configured on the router.
- Determine its RIP configuration.

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Create the various IP interfaces.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
interface create ip to-r6  address-netmask  160.1.1.1/16 port et.1.6
interface create ip to-r7  address-netmask  170.1.1.1/16 port et.1.7
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure a default route through 170.1.1.7
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ip add route default gateway 170.1.1.7
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure default routes to the 135.3.0.0 subnets reachable through
! R3.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ip add route 135.3.1.0/24 gateway 130.1.1.3
ip add route 135.3.2.0/24 gateway 130.1.1.3
ip add route 135.3.3.0/24 gateway 130.1.1.3
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure default routes to the other subnets reachable through R2.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ip add route 202.1.0.0/16 gateway 120.190.1.2
ip add route 160.1.5.0/24 gateway 120.190.1.2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! RIP Box Level Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
rip start
rip set default-metric 2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! RIP Interface Configuration. Create a RIP interfaces, and set
! their type to (version II, multicast).
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
rip add interface to-r41
rip add interface to-r42
rip add interface to-r6
rip set interface to-r41 version 2 type multicast
rip set interface to-r42 version 2 type multicast
rip set interface to-r6  version 2 type multicast
```

Importing a selected subset of routes from one of the RIP trusted gateways

Router R1 has several RIP peers. Router R41 has an interface on the network 10.51.0.0. By default, router R41 advertises network 10.51.0.0/16 in its RIP updates. Router R1 would like to import all routes except the 10.51.0.0/16 route from its peer R41.

1. Add the peer 140.1.1.41 to the list of trusted and source gateways.

```
rip add source-gateways 140.1.1.41
```

```
rip add trusted-gateways 140.1.1.41
```

2. Create a RIP import source with the gateway as 140.1.1.4 since we would like to import all routes except the 10.51.0.0/16 route from this gateway.

```
ip-router policy create rip-import-source ripImpSrc144 gateway 140.1.1.4
```

3. Create the Import-Policy, importing all routes except the 10.51.0.0/16 route from gateway 140.1.1.4

```
ip-router policy import source ripImpSrc144 network all
```

```
ip-router policy import source ripImpSrc144 network 10.51.0.0/16 restrict
```

Importing a selected subset of routes from all RIP peers accessible over a certain interface

Router R1 has several RIP peers. Router R41 has an interface on the network 10.51.0.0. By default, router R41 advertises network 10.51.0.0/16 in its RIP updates. Router R1 would like to import all routes except the 10.51.0.0/16 route from all its peer which are accessible over interface 140.1.1.1.

1. Create a RIP import source with the interface as 140.1.1.1, since we would like to import all routes except the 10.51.0.0/16 route from this interface.

```
ip-router policy create rip-import-source ripImpSrc140 interface 140.1.1.1
```

2. Create the Import-Policy importing all routes except the 10.51.0.0/16 route from interface 140.1.1.1

```
ip-router policy import source ripImpSrc140 network all
```

```
ip-router policy import source ripImpSrc140 network 10.51.0.0/16 restrict
```

Example 2: Importing from OSPF

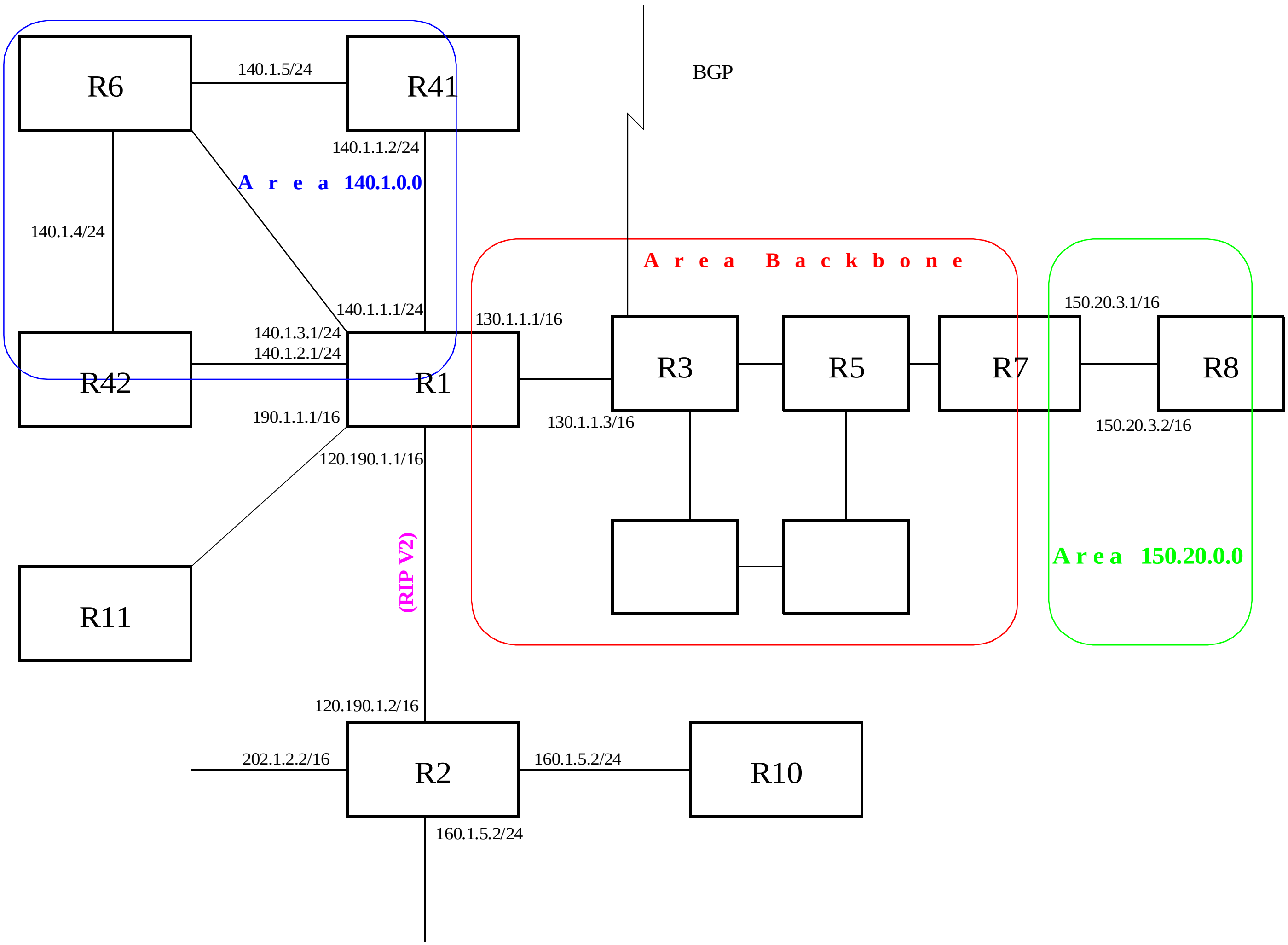
Due to the nature of OSPF, only the importation of ASE routes may be controlled. OSPF intra-and inter-area routes are always imported into the GIGAswitch/Router routing table with a preference of 10. If a tag is specified, the import clause will only apply to routes with the specified tag.

It is only possible to restrict the importation of OSPF ASE routes when functioning as an AS border router.

Like the other interior protocols, preference cannot be used to choose between OSPF ASE routes. That is done by the OSPF costs. Routes that are rejected by policy are stored in the table with a negative preference.

For all examples in this section, refer to the configuration shown in Figure 3 on page 6 - 25.

Figure 3: Exporting to OSPF



The following configuration commands for router R1:

- Determine the IP address for each interface
- Specify the static routes configured on the router
- Determine its OSPF configuration

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Create the various IP interfaces.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  140.1.3.1/24 port et.1.6
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure default routes to the other subnets reachable through R2.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.1.1.2
   ip add route 160.1.5.0/24 gateway 120.1.1.2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! OSPF Box Level Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf start
   ospf create area 140.1.0.0
   ospf create area backbone
   ospf set ase-defaults cost 4
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! OSPF Interface Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf add interface 140.1.1.1 to-area 140.1.0.0
   ospf add interface 140.1.2.1 to-area 140.1.0.0
   ospf add interface 140.1.3.1 to-area 140.1.0.0
   ospf add interface 130.1.1.1 to-area backbone
```

Importing a selected subset of OSPF-ASE routes

1. Create a OSPF import source so that only routes that have a tag of 100 are considered for importation.

```
ip-router policy create ospf-import-source ospfImpSrct100 tag 100
```

2. Create the Import-Policy importing all OSPF ASE routes with a tag of 100 except the default ASE route.

```
ip-router policy import source ospfImpSrct100 network all

ip-router policy import source ospfImpSrct100 network default restrict
```

Examples of Export Policies

Example 1: Exporting to RIP

Exporting to RIP is controlled by any of protocol, interface or gateway. If more than one is specified, they are processed from most general (protocol) to most specific (gateway).

It is not possible to set metrics for exporting RIP routes into RIP. Attempts to do this are silently ignored.

If no export policy is specified, RIP and interface routes are exported into RIP. If any policy is specified, the defaults are overridden; it is necessary to explicitly specify everything that should be exported.

RIP version 1 assumes that all subnets of the shared network have the same subnet mask so it is only able to propagate subnets of that network. RIP version 2 removes that restriction, and is capable of propagating all routes when not sending version 1 compatible updates.

To announce routes which specify a next hop of the loopback interface (i.e. static and internally generated default routes) via RIP, it is necessary to specify the metric at some level in the export policy. Just setting a default metric for RIP is not sufficient. This is a safeguard to verify that the announcement is intended.

For all examples in this section, refer to the configuration shown in Figure 2 on page 6 - 21.

The following configuration commands for router R1:

- Determine the IP address for each interface
- Specify the static routes configured on the router
- Determine its RIP configuration

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Create the various IP interfaces.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  160.1.1.1/16 port et.1.6
   interface create ip to-r7  address-netmask  170.1.1.1/16 port et.1.7
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Configure a default route through 170.1.1.7
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route default gateway 170.1.1.7
```

```
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Configure default routes to the 135.3.0.0 subnets reachable through
   ! R3.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 135.3.1.0/24 gateway 130.1.1.3
   ip add route 135.3.2.0/24 gateway 130.1.1.3
   ip add route 135.3.3.0/24 gateway 130.1.1.3
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Configure default routes to the other subnets reachable through R2.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.190.1.2
   ip add route 160.1.5.0/24 gateway 120.190.1.2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! RIP Box Level Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   rip start
   rip set default-metric 2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! RIP Interface Configuration. Create a RIP interfaces, and set
   ! their type to (version II, multicast).
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   rip add interface to-r41
   rip add interface to-r42
   rip add interface to-r6
   rip set interface to-r41 version 2 type multicast
   rip set interface to-r42 version 2 type multicast
   rip set interface to-r6  version 2 type multicast
```

Exporting a given static route to all RIP interfaces

Router R1 has several static routes, of which one is the default route. We would export this default route over all RIP interfaces.

1. Create a RIP export destination since we would like to export routes into RIP.

   ```
   ip-router policy create rip-export-destination ripExpDst
   ```

2. Create a Static export source since we would like to export static routes.

   ```
   ip-router policy create static-export-source statExpSrc
   ```

   As mentioned above, if no export policy is specified, RIP and interface routes are exported into RIP. If any policy is specified, the defaults are overridden; it is necessary to explicitly specify everything that should be exported.

   Since we would also like to export/redistribute RIP and direct routes into RIP, we would also create export-sources for those protocols.

3. Create a RIP export source since we would like to export RIP routes.

```
ip-router policy create rip-export-source ripExpSrc
```

4. Create a Direct export source since we would like to export direct/interface routes.

```
ip-router policy create direct-export-source directExpSrc
```

5. Create the export-policy redistributing the statically created default route, and all (RIP, Direct) routes into RIP.

```
ip-router policy export destination ripExpDst source statExpSrc network
     default
```

```
ip-router policy export destination ripExpDst source ripExpSrc network all
```

```
ip-router policy export destination ripExpDst source directExpSrc network all
```

Exporting a given static route to a specific RIP interface

In this case, router R1 would export/redistribute the default route over its interface 140.1.1.1 only.

1. Create a RIP export destination for interface with address 140.1.1.1, since we intend to change the rip export policy only for interface 140.1.1.1.

```
ip-router policy create rip-export-destination ripExpDst141 interface
     140.1.1.1
```

2. Create a static export source since we would like to export static routes.

```
ip-router policy create static-export-source statExpSrc
```

3. Create a RIP export source since we would like to export RIP routes.

```
ip-router policy create rip-export-source ripExpSrc
```

4. Create a Direct export source since we would like to export direct/interface routes.

```
ip-router policy create direct-export-source directExpSrc
```

5. Create the Export-Policy redistributing the statically created default route, and all (RIP, Direct) routes into RIP.

```
ip-router policy export destination ripExpDst141 source statExpSrc network
     default
```

```
ip-router policy export destination ripExpDst141 source ripExpSrc network all
```

```
ip-router policy export destination ripExpDst141 source directExpSrc network
     all
```

Exporting all static routes reachable over a given interface to a specific RIP-interface

In this case, router R1 would export/redistribute all static routes accessible through its interface 130.1.1.1 to its RIP-interface 140.1.1.1 only.

1. Create a RIP export destination for interface with address 140.1.1.1, since we intend to change the rip export policy for interface 140.1.1.1

```
ip-router policy create rip-export-destination ripExpDst141 interface
     140.1.1.1
```

2. Create a Static export source since we would like to export static routes.

```
ip-router policy create static-export-source statExpSrc130
interface 130.1.1.1
```

3. Create a RIP export source since we would like to export RIP routes.

```
ip-router policy create rip-export-source ripExpSrc
```

4. Create a Direct export source.

```
ip-router policy create direct-export-source directExpSrc
```

5. Create the Export-Policy, redistributing all static routes reachable over interface 130.1.1.1 and all (RIP, Direct) routes into RIP.

```
ip-router policy export destination ripExpDst141 source statExpSrc130 network
     all
```

```
ip-router policy export destination ripExpDst141 source ripExpSrc network all
```

```
ip-router policy export destination ripExpDst141 source directExpSrc network
     all
```

Exporting aggregate-routes into RIP

In the configuration shown in Figure 2 on page 6 - 21, suppose you decide to run RIP Version 1 on network 130.1.0.0/16, connecting routers R1 and R3. Router R1 desires to announce the 140.1.1.0/24 and 140.1.2.0/24 networks to router R3. RIP Version 1 does not carry any information about subnet masks in its packets. Thus it would not be possible to announce the subnets (140.1.1.0/24 and 140.1.2.0/24) into RIP Version 1 without aggregating them.

1. Create an Aggregate-Destination which represents the aggregate/summarized route.

```
ip-router policy create aggr-gen-dest aggrDst140 network 140.1.0.0/16
```

2. Create an Aggregate-Source which qualifies the source of the routes contributing to the aggregate. Since in this case, we do not care about the source of the contributing routes, we would specify the protocol as all.

```
ip-router policy create aggr-gen-source allAggrSrc protocol all
```

3. Create the aggregate/summarized route. This command binds the aggregated route with the contributing routes.

```
ip-router aggr-gen destination aggrDst140 source allAggrSrc network 140.1.1.0/
     24
```

```
ip-router aggr-gen destination aggrDst140 source allAggrSrc network 140.1.2.0/
     24
```

4. Create a RIP export destination for interface with address 130.1.1.1, since we intend to change the rip export policy only for interface 130.1.1.1.

```
ip-router policy create rip-export-destination ripExpDst130 interface
     130.1.1.1
```

5. Create a Aggregate export source since we would to export/redistribute an aggregate/summarized route.

   ```
   ip-router policy create aggr-export-source aggrExpSrc
   ```

6. Create a RIP export source since we would like to export RIP routes.

   ```
   ip-router policy create rip-export-source ripExpSrc
   ```

7. Create a Direct export source since we would like to export Direct routes.

   ```
   ip-router policy create direct-export-source directExpSrc
   ```

8. Create the Export-Policy redistributing all (RIP, Direct) routes and the aggregate route 140.1.0.0/16 into RIP.

```
ip-router policy export destination ripExpDst130 source aggrExpSrc network
     140.1.0.0/16

ip-router policy export destination ripExpDst130 source ripExpSrc network all

ip-router policy export destination ripExpDst130 source directExpSrc network
     all
```

Example 2: Exporting to OSPF

It is not possible to create OSPF intra- or inter-area routes by exporting routes from the GIGAswitch/Router routing table into OSPF. It is only possible to export from the GIGAswitch/Router routing table into OSPF ASE routes. It is also not possible to control the propagation of OSPF routes within the OSPF protocol.

There are two types of OSPF ASE routes: type 1 and type 2. The default type is specified by the **ospf set ase-defaults type 1/2** command. This may be overridden by a specification in the **ip-router policy create ospf-export-destination** command.

OSPF ASE routes also have the provision to carry a tag. This is an arbitrary 32-bit number that can be used on OSPF routers to filter routing information. The default tag is specified by the **ospf set ase-defaults tag** command. This may be overridden by a tag specified with the **ip-router policy create ospf-export-destination** command.

Interface routes are not automatically exported into OSPF. They have to be explicitly done.

For all examples in this section, refer to the configuration shown in Figure 3 on page 6 - 25.

The following configuration commands for router R1:

- Determine the IP address for each interface
- Specify the static routes configured on the router
- Determine its OSPF configuration

```
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Create the various IP interfaces.
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  interface create ip to-r2  address-netmask  120.190.1.1/16 port et.1.2
   interface create ip to-r3  address-netmask  130.1.1.1/16 port et.1.3
   interface create ip to-r41 address-netmask  140.1.1.1/24 port et.1.4
   interface create ip to-r42 address-netmask  140.1.2.1/24 port et.1.5
   interface create ip to-r6  address-netmask  140.1.3.1/24 port et.1.6
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! Configure default routes to the other subnets reachable through R2.
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ip add route 202.1.0.0/16 gateway 120.1.1.2
   ip add route 160.1.5.0/24 gateway 120.1.1.2
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! OSPF Box Level Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf start
   ospf create area 140.1.0.0
   ospf create area backbone
   ospf set ase-defaults cost 4
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ! OSPF Interface Configuration
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ospf add interface 140.1.1.1 to-area 140.1.0.0
   ospf add interface 140.1.2.1 to-area 140.1.0.0
   ospf add interface 140.1.3.1 to-area 140.1.0.0
   ospf add interface 130.1.1.1 to-area backbone
```

Exporting all interface and static routes to OSPF

Router R1 has several static routes. We would export these static routes as type-2 OSPF routes. The interface routes would redistributed as type 1 OSPF routes.

1. Create a OSPF export destination for type-1 routes since we would like to redistribute certain routes into OSPF as type 1 OSPF-ASE routes.

```
ip-router policy create ospf-export-destination ospfExpDstType1 type 1 metric
     1
```

2. Create a OSPF export destination for type-2 routes since we would like to redistribute certain routes into OSPF as type 2 OSPF-ASE routes.

```
ip-router policy create ospf-export-destination ospfExpDstType2 type 2 metric
     4
```

3. Create a Static export source since we would like to export static routes.

   ```
   ip-router policy create static-export-source statExpSrc
   ```

4. Create a Direct export source since we would like to export interface/direct routes.

   ```
   ip-router policy create direct-export-source directExpSrc
   ```

5. Create the Export-Policy for redistributing all interface routes and static routes into OSPF.

```
ip-router policy export destination ospfExpDstType1 source directExpSrc
     network all
```

```
ip-router policy export destination ospfExpDstType2 source statExpSrc network
     all
```

Exporting all RIP, interface, and static routes to OSPF

Note: Also export interface, static, RIP, OSPF, and OSPF-ASE routes into RIP.

In the configuration shown in Figure 3 on page 6 - 25, suppose if we decide to run RIP Version 2 on network 120.190.0.0/16, connecting routers R1 and R2.

We would like to redistribute these RIP routes as OSPF type-2 routes, and associate the tag 100 with them. Router R1 would also like to redistribute its static routes as type 2 OSPF routes. The interface routes would redistributed as type 1 OSPF routes.

Router R1 would like to redistribute its OSPF, OSPF-ASE, RIP, Static and Interface/ Direct routes into RIP.

1. Enable RIP on interface 120.190.1.1/16.

   ```
   rip add interface 120.190.1.1
   ```

   ```
   rip set interface 120.190.1.1 version 2 type multicast
   ```

2. Create a OSPF export destination for type-1 routes.

```
ip-router policy create ospf-export-destination ospfExpDstType1 type 1 metric
     1
```

3. Create a OSPF export destination for type-2 routes.

```
ip-router policy create ospf-export-destination ospfExpDstType2 type 2 metric
     4
```

4. Create a OSPF export destination for type-2 routes with a tag of 100.

```
ip-router policy create ospf-export-destination ospfExpDstType2t100 type 2 tag
     100 metric 4
```

5. Create a RIP export source.

```
ip-router policy export destination ripExpDst source ripExpSrc network all
```

6. Create a Static export source.

```
ip-router policy create static-export-source statExpSrc
```

7. Create a Direct export source.

```
ip-router policy create direct-export-source directExpSrc
```

8. Create the Export-Policy for redistributing all interface, RIP and static routes into OSPF.

```
ip-router policy export destination ospfExpDstType1 source directExpSrc
     network all
```

```
ip-router policy export destination ospfExpDstType2 source statExpSrc network
     all
```

```
ip-router policy export destination ospfExpDstType2t100 source ripExpSrc
     network all
```

9. Create a RIP export destination.

```
ip-router policy create rip-export-destination ripExpDst
```

10. Create OSPF export source.

```
ip-router policy create ospf-export-source ospfExpSrc type OSPF
```

11. Create OSPF-ASE export source.

```
ip-router policy create ospf-export-source ospfAseExpSrc type OSPF-ASE
```

12. Create the Export-Policy for redistributing all interface, RIP, static, OSPF and OSPF-ASE routes into RIP.

```
ip-router policy export destination ripExpDst source statExpSrc network all
```

```
ip-router policy export destination ripExpDst source ripExpSrc network all
```

```
ip-router policy export destination ripExpDst source directExpSrc network all
```

```
ip-router policy export destination ripExpDst source ospfExpSrc network all
```

```
ip-router policy export destination ripExpDst source ospfAseExpSrc network all
```

Chapter 7 Multicast Routing Configuration Guide

IP Multicast Overview

Multicast routing on the DIGITAL GIGAswitch/Router is supported through DVMRP and IGMP. IGMP is used to determine host membership on directly attached subnets. DVMRP is used to determine forwarding of multicast traffic between GIGAswitch/ Routers.

This chapter:

- Provides an overview of the GIGAswitch/Router's implementation of the Internet Group Management Protocol (IGMP)
- Provides an overview of the GIGAswitch/Router's implementation of the Distance Vector Multicast Routing Protocol (DVMRP)
- Discusses configuring DVMRP routing on the GIGAswitch/Router
- Discusses configuring IGMP on the GIGAswitch/Router.

IGMP Overview

The GIGAswitch/Router supports IGMP Version 2.0 as defined in RFC 2236. IGMP is run on a per-IP interface basis. An IP interface can be configured to run just IGMP and not DVMRP. Since multiple physical ports (VLANs) can be configured with the same IP interface on the GIGAswitch/Router, IGMP keeps track of multicast host members on a per-port basis. Ports belonging to an IP VLAN without any IGMP membership will not be forwarded any multicast traffic.

IGMP allows per-interface control of the host query interval and response time. Query interval defines the time between IGMP queries. Response time defines the time the GIGAswitch/Router will wait for host responses to IGMP queries.

DVMRP Overview

DVMRP is an IP multicast routing protocol. On the GIGAswitch/Router, DVMRP routing is implemented as specified in the draft-ietf-idmr-dvmrp-v3-06.txt file, which is an Internet Engineering Task Force (IETF) document. The GIGAswitch/Router's implementation of DVMRP supports the following:

- mtrace, which is a utility that tracks the multicast path from a source to a receiver.
- Generation identifiers, which are assigned to DVMRP whenever that protocol is

started on a router.

- Pruning, which is an operation DVMRP routers perform to exclude interfaces not in the shortest path tree.

DVMRP uses the Reverse Path Multicasting (RPM) algorithm to perform pruning.In RPM, a source network rather than a host is paired with a multicast group. RPM permits the GIGAswitch/Router to maintain multiple multicast groups.

On the GIGAswitch/Router, DVMRP can be configured on a per-interface basis. An interface does not have to run both DVMRP and IGMP. You can start and stop DVMRP independently from other routing protocols. IGMP starts and stops automatically with DVMRP.

To support backward compatibility on DVMRP interfaces, you can configure the router expire time and prune time on each GIGAswitch/Router DVMRP interface. This lets it work with older versions of DVMRP.

You can use threshold values and scopes to control internetwork traffic on each DVMRP interface. Threshold values determine whether traffic is either restricted or not restricted to a subnet, site, or region. Scopes define a set of multicast addresses of devices to which the GIGAswitch/Router can send DVMRP data. Scopes can include only addresses of devices on a company's internal network and cannot include addresses that require the GIGAswitch/Router to send DVMRP data on the internet.

You can also configure tunnels on GIGAswitch/Router DVMRP interfaces. A tunnel is used to send packets between routers separated by gateways that do not support multicast routing. A tunnel acts as a virtual network between two routers running DVMRP. A tunnel does not run IGMP.

Configuring IGMP

You configure IGMP on the GIGAswitch/Router by performing the following configuration tasks.

- Creating IP interfaces.
- Setting global parameters that will be used for all the interfaces on which DVMRP is enabled.
- Configuring IGMP on individual interfaces. You do so by enabling and disabling IGMP on interfaces and then setting IGMP parameters on the interfaces on which IGMP is enabled.

Configuring IGMP on an IP Interface

By default IGMP is disabled on the GIGAswitch/Router.

To enable IGMP on an interface, enter the following command in Configure mode:

Enable IGMP on an interface.	**igmp enable interface** *<ipAddr>*

Configuring IGMP Query Interval

You can configure the GIGAswitch/Router with a different IGMP Host Membership Query time interval. The interval you set applies to all ports on the GIGAswitch/Router. The default query time interval is 125 seconds.

To configure the IGMP host membership query time interval, enter the following command in Configure mode:

Configure the IGMP host membership query time interval.	**igmp set queryinterval** *<num>*

Configuring IGMP Response Wait Time

You can configure the GIGAswitch/Router with a wait time for IGMP Host Membership responses which is different from the default. The wait time you set then applies to all ports on the GIGAswitch/Router. The default response time is 10 seconds.

To configure the host response wait time, enter the following command in Configure mode:

Configure the IGMP host response wait time.	**igmp set responsetime** *<num>*

Configuring Per-Interface Control of IGMP Membership

You can configure the GIGAswitch/Router to control IGMP membership on a per-interface basis. An interface can be configured to be allowed or not allowed membership to a particular group.

To configure the per-interface membership control, enter the following commands in Configure mode:

Allow a host group membership to a specific group.	**igmp set interface** *<ip-addr>* **allowed-groups** *<ip-addr/subnet mask>*
Disallow a host group membership to a specific group.	**igmp set interface** *<ip-addr>* **not-allowed-groups** *<ip-addr/subnet mask>*

Configuring DVMRP

You configure DVMRP routing on the GIGAswitch/Router by performing the following DVMRP-configuration tasks.

- Creating IP interfaces.
- Setting global parameters that will be used for all the interfaces on which DVMRP is enabled.
- Configuring DVMRP on individual interfaces. You do so by enabling and disabling DVMRP on interfaces and then setting DVMRP parameters on the interfaces on which DVMRP is disabled.
- Defining DVMRP tunnels, which IP uses to send multicast traffic between two end points.

Starting and Stopping DVMRP

DVMRP is disabled by default on the GIGAswitch/Router.

To start or stop DVMRP, enter one of the following commands in Configure mode:

Start DVMRP.	**`dvmrp start`**
Stop DVMRP.	**`no dvmrp start`**

Configuring DVMRP on an Interface

DVMRP can be controlled/configured on per-interface basis. An interface does not have to run both DVMRP and IGMP together. DVMRP can be started or stopped IGMP starts and stops automatically with DVMRP.

To enable IGMP on an interface, enter the following command in the Configure mode:

Enable DVMRP on an interface.	**`dvmrp enable interface`** *`<ipAddr>`*

Configuring DVMRP Parameters

In order to support backward compatibility, DVMRP neighbor timeout and prune time can be configured on a per-interface basis. The default neighbor timeout is 35 seconds. The default prune time is 7200 seconds.

To configure neighbor timeout or prune time, enter one of the following commands in Configure mode:

Configure the DVMRP neighbor timeout.	**dvmrp set interface** *<ip-addr>* **neighbor-timeout***<number>*
Configure the DVMRP prune time.	**dvmrp set interface** *<ip-addr>* **prunetime** *<number>*

Configuring the DVMRP Routing Metric

You can configure the DVMRP routing metric associated with a set of destinations for DVMRP reports. The default metric is 1.

To configure the DVMRP routing metric, enter the following command in Configure mode:

Configure the DVMRP routing metric.	**dvmrp set interface** *<ip-addr>* **metric** *<number>*

Configuring DVMRP TTL and Scope

For control over internet traffic, per-interface control is allowed through Scopes and TTL thresholds.

The TTL value controls whether packets are forwarded from an interface. Conventional guidelines for assigning TTL values to a multicast application, and their corresponding GIGAswitch/Router setting for DVMRP threshold:

TTL = 1	Threshold = 1	Application restricted to subnet
TTL < 16	Threshold = 16	Application restricted to a site
TTL < 64	Threshold = 64	Application restricted to a region
TTL < 128	Threshold = 128	Application restricted to a continent
TTL = 255		Application not restricted

To configure the TTL Threshold, enter the following command in Configure mode:

Configure the TTL Threshold.	**dvmrp set interface** *<ip-addr>* **threshold** *<number>*

TTL thresholding is not always considered useful. There is another approach of a range of multicast addresses for “administrative” scoping. In other words, such addresses would be usable within a certain administrative scope, a corporate network, for instance, but would not be forwarded across the internet. The range from 239.0.0.0 through 239.255.255.255 is being reserved for administratively scoped applications. Any organization can currently assign this range of addresses and the packets will not be sent out of the organization. In addition, multiple scopes can be defined on per-interface basis.

To prevent the GIGAswitch/Router from forwarding any data destined to a scoped group on an interface, enter the following command in the Configure mode:

Configure the DVMRP scope.	**`dvmrp set interface`** *`<ip-addr>`* **`scope`** *`<ip-addr/mask>`*

Configuring a DVMRP Tunnel

The GIGAswitch/Router supports DVMRP tunnels to the MBONE (the multicast backbone of the Internet). You can configure a DVMRP tunnel on a router if the other end is running DVMRP. The GIGAswitch/Router then sends and receives multicast packets over the tunnel.

DVMRP tunnels need to be created before being enabled. Tunnels are recognized by the tunnel name. Once a DVMRP tunnel is created, you can enable DVMRP on the interface.

To configure a DVMRP tunnel, enter the following command in Configure mode:

Configure a DVMRP tunnel to MBONE.	**`dvmrp create tunnel`** *`<string>`* **`local`** *`<ip-addr>`* **`remote`** *`<ip-addr>`*

You can also control the rate of DVMRP traffic in a DVMRP tunnel. The default rate is 500 Kbps.

To control the rate of DVMRP traffic, enter the following command in Configure mode:

Configure the rate in a DVMRP tunnel.	**`dvmrp set interface`** *`<ip-addr>`* **`rate`** *`<number>`*

Monitoring IGMP and DVMRP

You can monitor IGMP and DVMRP information on the GIGAswitch/Router.

To display IGMP and DVMRP information, enter the following commands in the Enable mode.

Show all interfaces running DVMRP. Also shows the neighbors on each interface.	**dvmrp show interface**
Display DVMRP routing table.	**dvmrp show routes**
Shows all the interfaces and membership details running IGMP.	**igmp show interface**
Shows all IGMP group memberships on a port basis.	**igmp show memberships**
Show all IGMP timers.	**igmp show timers**
Show information about multicasts registered by IGMP.	**l2-tables show igmp-mcast-registration**
Show IGMP status on a VLAN.	**l2-tables show vlan-igmp-status**
Show all multicast Source, Group entries.	**mulitcast show cache**
Show all interfaces running multicast protocols (IGMP, DVMRP).	**multicast show interfaces**
Show all multicast routes.	**multicast show mroutes**

Configuration Examples

The following is a sample GIGAswitch/Router configuration for DVMRP and IGMP. Seven subnets are created. IGMP is enabled on 4 IP interfaces. The IGMP query interval is set to 30 seconds. DVMRP is enabled on 5 IP interfaces. IGMP is not running on “downstream” interfaces.

```
! Create VLANS.
!
vlan create upstream ip
vlan add ports et.5.3,et.5.4 to upstream
!
```

```
! Create IP intefaces
!
interface create ip mls15 address-netmask 172.1.1.10/24 port et.5.8
interface create ip company address-netmask 207.135.89.64/25 port
et.5.1
interface create ip test address-netmask 10.135.89.10/25 port et.1.8
interface create ip rip address-netmask 190.1.0.1 port et.1.4
interface create ip mbone address-netmask 207.135.122.11/29 port et.1.1
interface create ip downstream address-netmask 10.40.1.10/24 vlan
upstream
!
! Enable IGMP interfaces.
!
igmp enable interface 10.135.89.10
igmp enable interface 172.1.1.10
igmp enable interface 207.135.122.11
igmp enable interface 207.135.89.64
!
! Set IGMP Query Interval
!
igmp set queryinterval 30
!
! Enable DVMRP
!
dvmrp enable interface 10.135.89.10
dvmrp enable interface 172.1.1.10
dvmrp enable interface 207.135.122.11
dvmrp enable interface 207.135.89.64
dvmrp enable interface 10.40.1.10
!
! Set DVMRP parameters
!
dvmrp set interface 172.1.1.10 neighbor-timeout 200
!
! Start DVMRP
!
dvmrp start
```

Chapter 8 IPX Routing Configuration Guide

IPX Routing Overview

The Internetwork Packet Exchange (IPX) is a datagram connectionless protocol for the Novell NetWare environment. You can configure the DIGITAL GIGAswitch/Router for IPX routing and SAP. Routers interconnect different network segments and by definitions are network layer devices. Thus routers receive their instructions for forwarding a packet from one segment to another from a network layer protocol. IPX, with the help of RIP and SAP, perform these Network Layer Task. These tasks include addressing, routing, and switching information packets from one location to another on the internetwork.

IPX defines internetwork and intranode addressing schemes. IPX internetwork addressing is based on network numbers assigned to each network segment on a Novell NetWare internetwork. The IPX intranode address comes in the form of socket numbers. Because several processes are normally operating within a node, socket numbers provide a way for each process to distinguish itself.

The IPX packet consists of two parts: a 30-byte header and a data portion. The network node and socket addresses for both the destination and source are held within the IPX header.

RIP (Routing Information Protocol)

IPX routers use RIP to create and dynamically maintain a database of internetwork routing information. RIP allows a router to exchange routing information with a neighboring router. As a router becomes aware of any change in the internetwork layout, this information is immediately broadcast to any neighboring routers. Routers also send periodic RIP broadcast packets containing all routing information known to the router.

The GIGAswitch/Router uses IPX RIP to create and maintain a database of internetwork routing information. The GIGAswitch/Router's implementation of RIP allows the following exchanges of information:

- Workstations locate the fastest route to a network number by broadcasting a route request.
- Routers request routing information from other routers to update their own internal tables by broadcasting a route request.
- Routers respond to route requests from workstations and other routers.
- Routers perform periodic broadcasts to make sure that all other routers are aware of

the internetwork configuration.

- Routers perform broadcasting whenever they detect a change in the internetwork configurations.

GIGAswitch/Router's RIP implementation follows the guidelines given in Novell's *IPX RIP and SAP Router Specification Version 1.30* document.

SAP (Service Advertising Protocol)

SAP provides routers with a means of exchanging internetwork service information. Though SAP, servers advertise their services and addresses. Routers gather this information and share it with other routers. This allows routers to create and dynamically maintain a database of internetwork service information. SAP allows a router to exchange information with a neighboring SAP agent. As a router becomes aware of any change in the internetwork server layout, this information is immediately broadcast to any neighboring SAP agents. SAP broadcast packets containing all server information known to the router are also sent periodically.

The GIGAswitch/Router uses IPX SAP to create and maintain a database of internetwork service information. The GIGAswitch/Router’s implementation of SAP allows the following exchanges of information:

- Workstations locate the name and address of the nearest server of certain type
- Router’s request for the names and addresses of either all or certain type of servers
- Response to workstation or router’s request
- Periodic broadcast to make sure all other routers are aware of the internetwork configuration
- Perform broadcasting whenever they detect a change in the internetwork configurations

Configuring IPX RIP and SAP

This section provides an overview of configuring various IPX parameters and setting up IPX interfaces.

IPX RIP

On the GIGAswitch/Router, RIP automatically runs on all IPX interfaces. The GIGAswitch/Router will keep multiple routes to the same network having the lowest ticks and hop count. Static routes can be configured on the GIGAswitch/Router using the CLI’s **`ipx add route`** command. Through the use of RIP filters, the GIGAswitch/Router can control the acceptance and advertisement of networks per-interface.

IPX SAP

On the GIGAswitch/Router, SAP automatically runs on all the IPX interfaces. The GIGAswitch/Router will keep multiple SAP's having the lowest hop count. Static SAPs can be configured on the GIGAswitch/Router using the CLI's **ipx add sap** command. Through the use of SAP filters, the GIGAswitch/Router can control the acceptance and advertisements of services per-interface.

Creating IPX Interfaces

When you create IPX interfaces on the GIGAswitch/Router, you provide information about the interface (such as its name, output MAC encapsulation, and IPX address). You also enable or disable the interface and bind the interface to a single port or VLAN.

Note: Interfaces bound to a single port go down when the port goes down but interfaces bound to a VLAN remain up as long as at least one port in that VLAN remains active.

The procedure for creating an IPX interface depends on whether you are binding that interface to a single port or a VLAN. Separate discussions on the different procedures follow.

IPX Addresses

The IPX address is a 12-byte number divided into three parts. The first part is the 4-byte (8-character) IPX external network number. The second part is the 6-byte (12-character) node number. The third part is the 2-byte (4-character) socket number.

Configuring IPX Interfaces and Parameters

This section provides an overview of configuring various IPX parameters and setting up IPX interfaces.

Configuring IPX Addresses to Ports

You can configure one IPX interface directly to a physical port.

To configure an IPX interface to a port, enter one of the following commands in Configure mode:

Configure an IPX interface to a physical port.	**`interface create ipx`** *`<InterfaceName>`* **`address-mask`** *`<ipxAddr-mask>`* **`port`** *`<port>`*

Configuring IPX Interfaces for a VLAN

You can configure one IPX interface per VLAN.

To configure a VLAN with an IPX interface, enter the following command in Configure mode:

Create an IPX interface for a VLAN.	**interface create ipx** *<InterfaceName>* **address-mask** *<ipxAddr-mask>* **vlan** *<name>*

Specifying IPX Encapsulation Method

The DIGITAL GIGAswitch/Router supports two encapsulation types for IPX. You can configure encapsulation type on a per-interface basis.

- Ethernet II The standard ARPA Ethernet Version 2.0 encapsulation, which uses a 16-bit protocol type code (the default encapsulation method)
- 802.3 SNAP: SNAP IEEE 802.3 encapsulation, in which the type code becomes the frame length for the IEEE 802.2 LLC encapsulation (destination and source Service Access Points, and a control byte)
- 802.3: 802.3 encapsulation method used within Novell IPX environments
- 802.2: 802.2 encapsulation method used within Novell IPX environments

Configure Ethernet II encapsulation.	**interface create ipx** *<InterfaceName>* **output-mac-encapsulation ethernet_II**
Configure 802.3 SNAP encapsulation.	**interface create ipx** *<InterfaceName>* **output-mac-encapsulation ethernet_snap**
Configure 802.3 IPX encapsulation.	**interface create ipx** *<InterfaceName>* **output-mac-encapsulation ethernet_802.3**
Configure 802.2 IPX encapsulation.	**interface create ipx** *<InterfaceName>* **output-mac-encapsulation ethernet_802.2_ipx**

Configuring IPX Routing

By default, IPX routing is enabled on the GIGAswitch/Router.

Enabling IPX RIP

IPX RIP is enabled by default on the GIGAswitch/Router. You must first create an IPX interface or assign an IPX interface to a VLAN before RIP will start learning routes.

Enabling SAP

IPX SAP is enabled by default on the GIGAswitch/Router. You must first create an IPX interface or assign an IPX interface to a VLAN before SAP will start learning services.

Configuring Static Routes

In a Novell NetWare network, the GIGAswitch/Router uses RIP to determine the best paths for routing IPX. However, you can add static RIP routes to RIP routing table to explicitly specify a route.

To add a static RIP route, enter the following command in Configure mode:

Add a static RIP route.	**ipx add route** *<networkaddr> <nextrouter or network node> <metric> <ticks>*

Configuring Static SAP Table Entries

Servers in an IPX network use SAP to advertise services via broadcast packets. Services from servers are stored in the Server Information Table. If you want to have a service explicitly advertised with different hops then you will need to configure a static entry.

To add an entry into the Server Information Table, enter the following command in Configure mode:

Add a SAP table entry.	**ipx add sap** *<service type> <SrvcName> <node> <socket> <metric> <interface-network>*

Controlling Access to IPX Networks

To control access to IPX networks, you create access control lists and then apply them with filters to individual interfaces. The GIGAswitch/Router supports the following IPX access lists that you can use to filter various kinds of traffic:

- IPX access control list: Restrict traffic based on the source address, destination address, source socket, destination socket, source network mask or destination network mask.
- SAP access control list: Restricts advertisements or learning of SAP services. These lists are used for SAP filters. They can also be used for Get Nearest Server (GNS) replies.
- RIP access control list: Restricts advertisements or learning of networks.

Creating an IPX Access Control List

IPX access control lists control which IPX traffic is received from or sent to an interface based on source address, destination address, source socket, destination socket, source network mask or destination network mask. This is used to permit or deny traffic from one IPX end node to another.

To create an IPX access control list, perform the following task in the Configure mode:

Create an IPX access control list.	**acl** *<name>* **permit|deny ipx** *<SrcNetworkNode> <DstNetworkNode> <SrcSocket> <SrcNetMask> <DstSocket> <DstNetMask>*

Once an IPX access control list has been created, you must apply the access control list to an IPX interface. To apply an IPX access control list, enter the following command in Configure mode:

Apply an IPX access control list.	**acl** *<name>* **apply interface** *<InterfaceName>* **input|output [logging [on|off]]**

Creating an IPX SAP Access Control List

IPX SAP access control lists control which SAP services are available on a server. To create an IPX SAP access control list, perform the following task in the Configure mode:

Create an IPX SAP access control list.	**acl** *<name>* **permit|deny ipxsap** *<ServerNetworkNode> <ServiceType> <ServiceName>*

Once an IPX SAP access control list has been created, you must apply the access control list to an IPX interface. To apply an IPX SAP access control list, enter the following command in Configure mode:

Apply an IPX SAP access control list.	**acl** *<name>* **apply interface** *<InterfaceName>* **input|output [logging [on|off]]**

Creating an IPX RIP Access Control List

IPX RIP access control lists control which RIP updates are allowed. To create an IPX RIP access control list, perform the following task in the Configure mode:

Create an IPX RIP access control list.	**acl** *<name>* **permit|deny ipxrip** *<FromNetwork> <ToNetwork>*

Once an IPX RIP access control list has been created, you must apply the access control list to an IPX interface. To apply an IPX RIP access control list, enter the following command in Configure mode:

Apply an IPX RIP access control list.	**acl** *<name>* **apply interface** *<InterfaceName>* **input|output [logging [on|off]**

Monitoring an IPX Network

The GIGAswitch/Router reports IPX interface information and RIP or SAP routing information.

To display IPX information, enter the following command in Enable mode:

Show a RIP entry in the IPX RIP table.	**ipx find rip** *<DstNetwork>*
Show a SAP entry in the IPX SAP table.	**ipx find sap** *<type> <ServiceType> <Service-Name> <ServerNetwork>*
Show IPX interface information.	**ipx show interfaces** *<interface-name>*
Show IPX RIP table.	**ipx show tables rip**
Show IPX routing table.	**ipx show tables routing**
Show IPX SAP table.	**ipx show tables sap**

Configuration Examples

This example performs the following configuration:

- Creates IPX interfaces
- Adds static RIP routes
- Adds static SAP entries7.pdf.zip
- Adds a RIP access list
- Adds a SAP access list

```
! Create interface ipx1 with ipx address AAAAAAAA
interface create ipx ipx1 address AAAAAAAA port et.1.1 output-
mac-encapsulation ethernet_802.2_IPX
!
! Create interface ipx2 with ipx address BBBBBBBB
interface create ipx ipx2 address BBBBBBBB port et.1.2 output-
mac-encapsulation ethernet_802.3
!
!Add static route to network 9
ipx add route 9 BBBBBBBB.01:02:03:04:05:06 1 1
!
!Add static sap
ipx add sap 0004 FILESERVER1 9.03:04:05:06:07:08 452 1 AAAAAAAA
!
!RIP Access List
acl 100 deny ipxrip 1 2
!
!RIP inbound filter
acl 100 apply interface ipx1 input
!
!SAP Access List
acl 200 deny ipxsap A.01:03:05:07:02:03 0004 FILESERVER2
!
!SAP outbound filter to interface ipx2
acl 200 apply interface ipx2 output
```

Chapter 9 Security Configuration Guide

Security Overview

The DIGITAL GIGAswitch/Router provides security features that help control access to the GIGAswitch/Router and filter traffic going through the GIGAswitch/Router. Access to the GIGAswitch/Router can be controlled by:

- Enabling TACACS
- Login authentication

Traffic filtering on the GIGAswitch/Router enables:

- Layer-2 security filters - Perform filtering on source or destination MAC addresses.
- Layer3/4 Access Control Lists - Perform filtering on source or destination IP address, source or destination TCP/UDP port, TOS or protocol type for IP traffic. Perform filtering on source or destination IPX address, or source or destination IPX socket. Perform access control to services provided on the GIGAswitch/Router, for example, Telnet server and HTTP server.

Configuring GIGAswitch/Router Access Security

Configuring TACACS

Enable mode access to the GIGAswitch/Router can be made secure by enabling a Terminal Access Controller Access Control System (TACACS) client. Without TACACS enabled, only local password authentication is performed on the GIGAswitch/Router. The TACACS client provides user name and password authentication for Enable mode. A TACACS server responds to the GIGAswitch/Router TACACS client to provide authentication.

You can configure up to five TACACS server targets on the GIGAswitch/Router. A timeout is set to tell the GIGAswitch/Router how long to wait for a response from TACACS servers.

To configure TACACS security, enter the following commands in the Configure mode:

Specify a TACACS server.	**`tacacs set host`** *`<hostname or IP-addr>`*
Set the TACACS time to wait for a TACACS server reply.	**`tacacs set timeout`** *`<number>`*

Determine TACACS action if no server responds.	**tacacs set last-resort password|succeed**
Enable TACACS.	**tacacs enable**

Monitoring TACACS

You can monitor TACACS configuration and statistics within the GIGAswitch/Router.

To monitor TACACS, enter the following commands in Enable mode:

Show TACACS server statistics.	**tacacs show stats**
Show all TACACS parameters.	**tacacs show all**

Configuring Passwords

The GIGAswitch/Router provides password authentication for accessing the User and Enable modes. If TACACS is not enabled on the GIGAswitch/Router, only local password authentication is performed.

To configure GIGAswitch/Router passwords, enter the following commands in Configure mode:

Set User mode password.	**system set password login** *<string>*
Set Enable mode password.	**system set password enable** *<string>*

L2 Security Filters

Layer-2 security filters on the GIGAswitch/Router allow you to configure ports to filter specific MAC addresses. When defining a Layer-2 security filter, you specify to which ports you want the filter to apply. You can specify the following security filters:

- Address filters

  These filters block traffic based on the frame's source MAC address, destination MAC address, or both source and destination MAC addresses in flow bridging mode. Address filters are always configured and applied to the input port.

- Port-to-address lock filters

  These filters prohibit a user connected to a locked port or set of ports from using another port.

- Static entry filters

  These filters allow or force traffic to go to a set of destination ports based on a frame's source MAC address, destination MAC address, or both source and destination MAC addresses in flow bridging mode. Static entries are always configured and applied at the input port.

- Secure port filters

  A secure filter shuts down access to the GIGAswitch/Router based on MAC addresses. All packets received by a port are dropped. When combined with static entries, however, these filters can be used to drop all received traffic but allow some frames to go through.

Configuring Layer-2 Address Filters

If you want to control access to a source or destination on a per-MAC address basis, you can configure an address filter. Address filters are always configured and applied to the input port. You can set address filters on the following:

- A source MAC address, which filters out any frame coming from a specific source MAC address.
- A destination MAC address, which filters out any frame destined to specific destination MAC address.
- A flow, which filters out any frame coming from a specific source MAC address that is also destined to a specific destination MAC address.

To configure Layer-2 address filters, enter the following commands in Configure mode:

Configure a source MAC based address filter.	**filters add address-filter name** *<name>* **source-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>*
Configure a destination MAC based address filter.	**filters add address-filter name** *<name>* **dest-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>*

Configure a Layer-2 flow address filter.	**filters add address-filter name** *<name>* **source-mac** *<MACaddr>* **dest-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>*

Configuring Layer-2 Port-to-Address Lock Filters

Port address lock filters allow you to bind or “lock” specific source MAC addresses to a port or set of ports. Once a port is locked, only the specified source MAC address is allowed to connect to the locked port and the specified source MAC address is not allowed to connect to any other ports.

To configure Layer-2 port address lock filters, enter the following commands in Configure mode:

Configure a port address lock filter.	**filters add port-address-lock name** *<name>* **source-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>*

Configuring Layer-2 Static Entry Filters

Static entry filters allow or force traffic to go to a set of destination ports based on a frame's source MAC address, destination MAC address, or both source and destination MAC addresses in flow bridging mode. Static entries are always configured and applied at the input port. You can set the following static entry filters:

- Source static entry, which specifies that any frame coming from source MAC address will be allowed or disallowed to go to a set of ports
- Destination static entry, which specifies that any frame destined to a specific destination MAC address will be allowed, disallowed, or forced to go to a set of ports
- Flow static entry, which specifies that any frame coming from a specific source MAC address that is destined to specific destination MAC address will be allowed, disallowed, or forced to go to a set of ports

To configure Layer-2 static entry filters, enter the following commands in Configure mode:

Configure a source static entry filter.	**filters add static-entry name** *<name>* **restriction allow|disallow|force source-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>* **out-port-list** *<port-list>*

Configure a destination static entry filter.	**filters add static-entry name** *<name>* **restriction allow|disallow|force dest-mac** *<MACaddr>* **vlan** *<VLAN-num>* **in-port-list** *<port-list>***out-port-list** *<port-list>*

Configuring Layer-2 Secure Port Filters

Secure port filters block access to a specified port. You can use a secure port filter by itself to secure unused ports. Secure port filters can be configured as source or destination port filters. A secure port filter applied to a source port forces all incoming packets to be dropped on a port. A secure port filter applied to a destination port prevents packets from going out a certain port.

You can combine secure port filters with static entries in the following ways:

- Combine a source secure port filter with a source static entry to drop all received traffic but allow any frame coming from specific source MAC address to go through
- Combine a source secure port filter with a flow static entry to drop all received traffic but allow any frame coming from a specific source MAC address that is destined to specific destination MAC address to go through
- Combine a destination secure port with a destination static entry to drop all received traffic but allow any frame destined to specific destination MAC address go through
- Combine a destination secure port filter with a flow static entry to drop all received traffic but allow any frame coming from specific source MAC address that is destined to specific destination MAC address to go through

To configure Layer-2 secure port filters, enter the following commands in Configure mode:

Configure a source secure port filter.	**filters add secure-port name** *<name>* **direction source vlan** *<VLAN-num>* **in-port-list** *<port-list>*
Configure a destination secure port filter.	**filters add secure-port name** *<name>* **direction destination vlan** *<VLAN-num>* **in-port-list** *<port-list>*

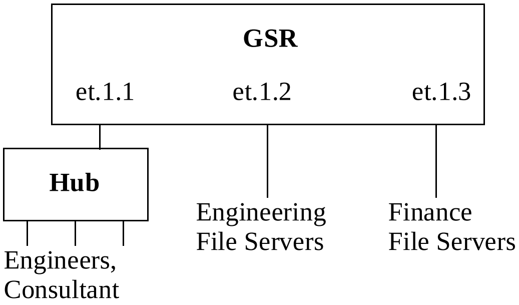
Monitoring Layer-2 Security Filters

The GIGAswitch/Router provides display of Layer-2 security filter configurations contained in the routing table.

To display security filter information, enter the following commands in Enable mode.

Show address filters.	**filters show address-filter** **[all-source|all-destination|all-flow]** **[source-mac** *<MACaddr>* **dest-mac** *<MACaddr>***]** **[ports** *<port-list>***]** **[vlan** *<VLAN-num>***]**
Show port address lock filters.	**filters show port-address-lock ports** **[ports** *<port-list>***]** **[vlan** *<VLAN-num>***] [source-mac** *<MACaddr>***]**
Show secure port filters.	**filters show secure-port**
Show static entry filters.	**filters show static-entry** **[all-source|all-destination|all-flow]** **ports** *<port-list>* **vlan** *<VLAN-num>* **[source-mac** *<MACaddr>* **dest-mac** *<MACaddr>***]**

Layer-2 Filter Examples



Example 1: Address Filters

Source filter: The consultant is not allowed to access any file servers. The consultant is only allowed to interact with the engineers on the same Ethernet segment – port et.1.1. All traffic coming from the consultant’s MAC address will be dropped.

```
filters add address-filter name consultant source-mac
001122:334455 vlan 1 in-port-list et.1.1
```

Destination filter: No one from the engineering group (port et.1.1) should be allowed to access the finance server. All traffic destined to the finance server's MAC will be dropped.

```
filters add address-filter name finance dest-mac AABBCC:DDEEFF
vlan 1 in-port-list et.1.1
```

Flow filter: Only the consultant is restricted access to one of the finance file servers. Note that port et.1.1 should be operating in flow-bridging mode for this filter to work.

```
filters add address-filter name consult-to-finance source-mac
001122:334455 dest-mac AABBCC:DDEEFF vlan 1 in-port-list et.1.1
```

Static Entries Example:

Source static entry: The consultant is only allowed to access the engineering file servers on port et.1.2.

```
filters add static-entry name consultant source-mac
001122:334455 vlan 1 in-port-list et.1.1 out-port-list et.1.2
restriction allow
```

Destination static entry: Restrict "login multicasts" originating from the engineering segment (port et.1.1) from reaching the finance servers.

```
filters add static-entry name login-mcasts dest-mac
010000:334455 vlan 1 in-port-list et.1.1 out-port-list et.1.3
restriction disallow
```

or

```
filters add static-entry name login-mcasts dest-mac
010000:334455 vlan 1 in-port-list et.1.1 out-port-list et.1.2
restriction allow
```

Flow static entry: Restrict "login multicasts" originating from the consultant from reaching the finance servers.

```
filters add static-entry name consult-to-mcasts source-mac
001122:334455  dest-mac 010000:334455 vlan 1 in-port-list et.1.1
out-port-list et.1.3 restriction disallow
```

Port-to-address Lock Examples:

You have configured some filters for the consultant on port et.1.1 If the consultant plugs his laptop into a different port, he will bypass the filters. To lock him to port et.1.1, use the following command:

```
filters add port-address-lock name consultant source-mac
001122:334455 vlan 1 in-port-list et.1.1
```

Note: If the consultant’s MAC is detected on a different port, all of its traffic will be blocked.

Example 2 : Secure Ports

Source secure port: To block all engineers on port 1 from accessing all other ports, enter the following command:

```
filters add secure-port name engineers direction source vlan 1
in-port-list et.1.1
```

To allow ONLY the engineering manager access to the engineering servers, you must "punch" a hole through the secure-port wall. A "source static-entry" overrides a "source secure port".

```
filters add static-entry name eng-mgr source-mac 080060:123456
vlan 1 in-port-list et.1.1 out-port-list et.1.2 restriction allow
```

Destination secure port: To block access to all file servers on all ports from port et.1.1 use the following command:

```
filters add secure-port name engineers direction dest vlan 1 in-
port-list et.1.1
```

To allow all engineers access to the engineering servers, you must "punch" a hole through the secure-port wall. A "dest static-entry" overrides a "dest secure port".

```
filters add static-entry name eng-server dest-mac 080060:abcdef
vlan 1 in-port-list et.1.1 out-port-list et.1.2 restriction allow
```

L3 Access Control Lists (ACLs)

Traffic Filters at Layer-3 and 4 (Access Control List)

Access Control Lists (ACLs) allow you to restrict Layer-3/4 traffic going through the router. Each ACL or each list consists of one or more rules describing a particular type of IP or IPX traffic. An ACL can be simple, consisting of only one rule, or complicated with many rules. Each rule tells the router to either permit or deny the packet that matches the rule's packet description.

The Anatomy of an ACL rule

Each ACL is identified by a name. The name can be a meaningful string, such as *denyftp* or *noweb* or it can be a number such as 100 or 101.

Each rule has an action, that is, to permit or to deny the packet if a packet satisfies the criterion defined by the rule.

A criterion describes one or more characteristics about a packet. In an ACL rule, these characteristics are described as fields of a rule. Not all characteristics (fields) of a packet (rule) need to be specified. If a particular field is not specified, it is treated as a wildcard or "don't care" condition. However, if a field is specified, that particular field will be matched against the packet. Each protocol can have a number of different fields to match. For example, TCP can use socket port numbers while IPX can use a network node address to define a rule. For IP, TCP and UDP ACLs, the following fields can be specified:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Type of Service (TOS)

For IPX ACLs, the following fields can be specified:

- Source network address
- Destination network address
- Source IPX socket
- Destination IPX socket

When defining an ACL rule, each field in the rule is position sensitive. For example, for TCP, the source address must be followed by the destination address, followed by the source socket and the destination socket and so on. For example, the following describes the syntax of a TCP ACL:

```
acl name permit tcp source-addr dest-addr source-port dest-port tos
```

Not all the fields are required. If a field is not specified, it is treated as don't care. However, since each field is position sensitive, it may be necessary to “skip” some fields in order to specify a value for another field. To skip a field, the keyword **any** is used. For example, the following rule denies SMTP traffic between any two hosts:

```
acl nosmtp deny tcp any any smtp smtp
```

Note that in the above example, the tos field (Type of Service) is not specified and is treated as don't care. The keyword any is needed only to skip a don't care field in order to explicitly specify another field that is further down in the rule. If there are no other fields to specify, the keyword any is not really needed. For example, the following ACL permits all IP traffic to go through:

```
acl yesip permit ip
```

The Ordering of ACL rules

For an ACL with multiple rules, the ordering of the rules is very important. When the router looks at an ACL to determine whether a packet should be forwarded or not, it goes through each rule in the ACL sequentially. When the router finds a rule that matches the packet, all subsequent rules are ignored. That is, a first match algorithm is used. The action defined by this ACL, to permit or deny, is used to forward or drop the packet. There are no hidden or implied ordering of these rules. Nor is there precedence attached to each field. The router simply goes down the list, one rule at a time until there is a match. Consequently, rules that are more specific (i.e. with more details) should always be listed ahead of rules that are less specific. For example, the following ACL permits all TCP traffic except those from subnet 10.2.0.0/16:

```
acl 101 deny tcp 10.2.0.0/16 any any any
acl 101 permit tcp any any any any
```

When a TCP packet comes from subnet 10.2.0.0/16, it finds a match with the first rule. This causes the packet to be dropped. A TCP packet coming from other subnets will not match the first rule. Instead, it matches the second rule which allows the packet to go through.

If you were to reverse the order of the two rules:

```
acl 101 permit tcp any any any any
acl 101 deny tcp 10.2.0.0/16 any any any
```

then all TCP packets will be allowed to go through, including traffic from subnet 10.2.0.0/16. This is because TCP traffic coming from 10.2.0.0/16 will match the first rule and be allowed to go. The second rule will not looked at since the first match determines the action on the packet.

Implicit Deny Rule

At the end of each ACL, the system automatically appends an implicit deny rule. This implicit deny rule denies all traffic. For a packet that doesn't match any of the user specified rules, the implicit deny rule acts as a catch all rule. All packets match correctly with this rule. The default behavior for a packet that doesn't match any rules in an ACL can be either to permit or to deny. The GIGAswitch/Router chooses to deny a packet as the default behavior. This is done for security reasons. If an ACL is misconfigured and a packet that should be allowed to go through is now blocked because of the implicit deny rule, the worse that could happen is inconvenience. On the other hand, if a packet that should not be allowed to go through is instead sent through, there is now a security breach. Basically, the implicit deny rule is the last line of defense against accidental mis-configuration of ACLs that could result in a security breach.

To describe how the implicit deny rule is used, considering the following example. Suppose someone created the following ACL:

```
acl 101 permit ip 1.2.3.4/24
acl 101 permit ip 4.3.2.1/24 any nntp
```

With the implicit deny rule, this ACL actually has three rules:

```
acl 101 permit ip 1.2.3.4/24 any any any
acl 101 permit ip 4.3.2.1/24 any nntp any
acl 101 deny any any any any any
```

If a packet comes in and doesn't match the first two rules, the packet will be dropped. This is because the third rule (implicit deny) will match all packets.

Although the implicit deny rule seems obvious in the above example, this is not always the case. For example, consider the following ACL rule:

```
acl 102 deny ip 10.1.20.0/24 any any any
```

If a packet comes in from a network other than 10.1.20.0/24, one might expect the packet to go through because it doesn't match the first rule. However, that is not the case because of the implicit deny rule. With the implicit deny rule attached, the rule looks like this:

```
acl 102 deny ip 10.1.20.0/24 any any any
acl 102 deny any any any any any
```

A packet coming from 10.1.20.0/24 will not match the first rule, but will match the implicit deny rule. As a result, no packets will be allowed to go through. Rule 1 is simply a subset of Rule 2. To allow packets from subnets other than 10.1.20.0/24 to go through, the administrator must explicitly define a rule to permit other packets to go through.

To fix the above example and let packets from other subnets enter the router, one must add a new rule to permit packets to go through:

```
acl 101 deny ip 10.1.20.0/24 any any any
acl 101 permit ip
acl 101 deny any any any any any
```

The second rule will forward all packets that are not denied by the first rule.

Due to the nature of the implicit deny rule, when creating an ACL, one should take the approach where a firewall is elected to deny all traffic. “Holes” are then punched into the firewall to permit specific types of traffic, for example, traffic from a specific subnet or traffic from a specific application.

Applying ACLs to Interfaces

Defining an ACL specifies what sort of traffic to permit or deny. However, an ACL has no effect unless it is applied to an interface. An ACL can be applied to examine either inbound or outbound traffic. Inbound traffic is traffic coming into the router. Outbound traffic is traffic going out of the router. For each interface, only one ACL can be applied for the same protocol in the same direction. For example, you cannot apply two or more IP ACLs to the same interface in the inbound direction. You can apply two ACLs to the same interface if one is for inbound traffic and one is for outbound trafic, but not in the same direction. However, this restriction does not prevent you from specifying many rules in an ACL. You just have to put all of these rules into one ACL and apply it to an interface.

When a packet comes into a router at an interface where an inbound ACL is applied, the router compares the packet with the rules specified by that ACL. If it is permitted, the packet is allowed into the router. If not, the packet is dropped. If that packet is to be forwarded to go out of another interface (that is, the packet is to be routed) then a second ACL check is possible. At the output interface, if an outbound ACL is applied, the packet will be compared with the rules specified in this outbound ACL. Consequently, it is possible for a packet to go through two separate checks, once at the inbound interface and once more at the outbound interface.

In general, you should try to apply ACLs at the inbound interfaces instead of the outbound interfaces. If a packet is to be denied, you want to drop the packet as early as possible, at the inbound interface. Otherwise, the router will have to process the packet, determine where the packet should go only to find out that the packet should be dropped at the outbound interface. In some cases, however, it may not be simple or possible for the administrator to know ahead of time that a packet should be dropped at the inbound interface. Nonetheless, for performance reason, whenever possible, one should create and apply an ACL to the inbound interface.

Applying ACLs to Services

ACLs can also be created to permit or deny access to system services provided by the router; for example, HTTP server or Telnet server. This type of ACL is known as a Service ACL. By definition, a Service ACL is for controlling inbound packets to a service on the router. For example, you can grant Telnet server access from a few specific hosts or deny Web server access from a particular subnet. It is true that one can do the same thing with ordinary ACLs and apply them to all interfaces. However, the Service ACL is created specifically to control access to some of the services on the router. As a result, the syntax of a Service ACL is much simpler than that of the ordinary ACL.

Note: If a service does not have an ACL applied then that service is accessible to everyone. To control access to a service, an ACL must be used.

ACL Logging

To see whether incoming packets are permitted or denied because of an ACL, one can enable ACL Logging when applying the ACL. When ACL Logging is turned on, the router prints out a message on the console about whether a packet is forwarded or dropped. If you have a Syslog server configured for the GIGAswitch/Router then the same information will also be sent to the Syslog server.

Before enabling ACL Logging, one should consider its impact on performance. With ACL Logging enabled, the router prints out a message at the console before the packet is actually forwarded or dropped. Even if the console is connected to the router at a high baud rate, the delay caused by the console message is still significant. This can get worse if the console is connected at a low baud rate, for example, 1200 baud. Furthermore, if a Syslog server is configured then a Syslog packet must also be sent to the Syslog server, creating additional delay. Therefore, one should consider the potential performance impact before turning on ACL Logging.

Maintaining ACLs offline using TFTP or RCP

The GIGAswitch/Router provides two mechanisms to maintain and manipulate ACLs. The traditional method used by some of the other popular routers require the use of TFTP or RCP. With this mechanism, the administrator is encouraged to create and modify ACLs on a remote host. The administrator can use his or her favorite editor to edit, delete, replace or reorder ACL rules in a file. Once the changes are made, the administrator can then download the ACLs to the router using TFTP or RCP and make them take effect on the running system.

The following example describes how one can use TFTP to help maintain ACLs on the GIGAswitch/Router. Suppose the following ACL commands are stored in a file on some hosts:

```
no acl *
acl 101 deny tcp 10.11.0.0/16 10.12.0.0/16
acl 101 permit tcp 10.11.0.0 any
acl 101 apply interface gsr1010 input
```

The first command, **`no acl *`**, negates all commands that start with the keyword, “acl”. This tells the router to remove the application and the definition of any ACL. The administrator can be more selective if he or she wants to remove only ACL commands related to, for instance, ACL 101 by saying, **`no acl 101 *`**. The negation of all related ACL commands is important because it removes any potential confusion caused by the addition of new ACL rules to existing rules. Basically, the **`no acl`** command cleans up the system for the new ACL rules.

Once the negation command is executed, the second and the third commands proceed to redefine ACL 101. The final command applies the ACL to interface gsr1010.

If the changes are accessible from a TFTP server, one can download and make the changes take effect by issuing commands like the following:

```
copy tftp://10.1.1.12/config/acl.changes to scratchpad
copy scratchpad to active
```

The first *copy* command downloads the file acl.changes from a TFTP server and puts the commands into the temporary configuration area, scratchpad. The administrator can re-examine the changes if necessary before committing the changes to the running system. The second copy command make the changes take effect by copying from the scratchpad to the active running system.

If the administrator needs to re-order or modify the ACL rules, one must make the changes in the acl.changes file on the remote host, download the changes and make them effective again.

Maintaining ACLs using the ACL Editor

In addition to the traditional method of maintaining ACLs using TFTP or RCP, the GIGAswitch/Router provides a simpler and more user-friendly mechanism to maintain ACL: the ACL Editor.

The ACL Editor can only be accessed within Configure mode using the **`acl-edit`** command. You can specify the ACL you want to edit by specifying its name together with the **`acl-edit`** command. For example, to edit ACL "101", you issue the command **`acl-edit 101`**. The only restriction is that when you edit a particular ACL, you cannot add rules for a different ACL. You can only add new rules for the ACL that you are currently editing. When the editing session is over, that is, when you are done making changes to the ACL, you can save the changes and make them take effect immediately. Within the ACL editor, you can add new rules (**`add`** command), delete existing rules (**`delete`** command) and re-order the rules (**`move`** command). To save the changes, use the **`save`** command or simply exit the editor.

If you edit and save changes to an ACL that is currently being used or applied to an interface, the changes will take effect immediately. There is no need to remove the ACL from the interface before making changes and re-apply after changes are made. The whole process is automatic.

Configuring ACL

To configure an ACL, perform the following tasks:

1. Determine the access control criteria you want to impose on traffic going to or through the router.
2. Determine where (which interface) you want to set up these controls.

Defining an IP ACL

To define an IP ACL, perform the following in the Configure mode:

Define an IP ACL.	**acl** *<name>* **permit|deny** **ip|tcp|udp|icmp|igmp** *<srcaddr/mask>***|any** *<dstaddr/mask>***|any** **Note:** Additional fields depend on the protocol type you select.

Defining an IPX ACL

To define an IPX ACL, perform the following in the Configure mode:

Define an IPX ACL.	**acl** *<name>* **permit|deny ipx|ipxrip|ipxsap** **Note:** Additional fields depend on the protocol type you select.

Applying an ACL to an Interface

To apply an ACL to an interface, perform the following in the Configure mode:

Apply ACL to an interface.	**acl** *<name>* **apply interface** *<Interface-Name>* **input|output [logging [on|off]]**

Applying an ACL to a Service

To apply an ACL to a service, perform the following in the Configure mode:

Apply ACL to a service.	**acl** *<name>* **apply service** *<ServiceName>* **[logging [on|off]]**

Editing an ACL with the ACL Editor

To edit an ACL, perform the following in the Configure mode:

Edit an ACL using the ACL Editor	**acl-edit <***aclname>*

Monitoring Access Control Lists

The GIGAswitch/Router provides display of ACL configurations contained in the system.

To display ACL information, enter the following command in Enable mode.

Show all ACLs.	**`acl show all`**
Show a specific ACL.	**`acl show aclname`** *`<Name>`* **`| all`**
Show an ACL on a specific interface.	**`acl show interface`** *`<Name>`*
Show ACLs on all IP interfaces.	**`acl show interface all-ip`**
Show ACLs on all IPX interfaces.	**`acl show interface all-ipx`**
Show static entry filters.	**`acl show service`**

Chapter 10 QoS Configuration Guide

QoS and L2/L3/L4 Flow Overview

The DIGITAL GIGAswitch/Router allows network managers to identify traffic and set Quality of Service (QoS) policies without compromising wire speed performance. The GIGAswitch/Router can guarantee bandwidth on an application by application basis, thus accommodating high-priority traffic even during peak periods of usage. QoS policies can be broad enough to encompass all the applications in the network, or relate specifically to a single host-to-host application flow.

Within the GIGAswitch/Router, QoS policies are used to classify Layer-2, Layer-3, and Layer-4 traffic into the following priorities:

- Control
- High
- Medium
- Low

By assigning priorities to network traffic, you can ensure that critical traffic will reach its destination even if the exit ports for the traffic are experiencing greater-than-maximum utilization.

Layer-2, 3, 4 Flow Specification

For Layer-2 traffic, you can define a flow based on the MAC packet headers.

- The MAC fields are source MAC address, destination MAC address and VLAN IDs. A list of incoming ports can also be specified

For Layer-3 (IP and IPX) traffic, you can define “flows”, blueprints or templates of IP and IPX packet headers.

- The IP fields are source IP address, destination IP address, UDP/TCP source port, UDP/TCP destination port, TOS (Type of Service), transport protocol (TCP or UDP), and a list of incoming interfaces
- The IPX fields are source network, source node, destination network, destination node, source port, destination port, and a list of incoming interfaces

The flows specify the contents of these fields. If you do not enter a value for a field, a wildcard value (all values acceptable) is assumed for the field.

Precedence for Layer-3 Flows

A precedence from 1 - 7 is associated with each field in a flow. The GIGAswitch/Router uses the precedence value associated with the fields to break ties if packets match more than one flow. The highest precedence is 1 and the lowest is 7. Here is the default precedence of the fields:

- IP - destination port (1), destination address (2), source port (3), source IP address (4), TOS (5), interface (6), protocol (7)
- IPX - destination network (1), source network (2), destination node (3), source node (4), destination port (5), source port (6), interface (7)

Use the **`qos precedence ip`** and **`qos precedence ipx`** commands to change the default precedence.

GIGAswitch/Router Queuing Policies

You can use one of two queuing policies on the GIGAswitch/Router:

- **strict priority** – assures the higher priorities of throughput but at the expense of lower priorities. For example, during heavy loads, low-priority traffic can be dropped to preserve throughput of control-priority traffic, and so on.
- **weighted fair queuing** – distributes priority throughput among the four priorities (control, high, medium, and low) based on percentages.

The GIGAswitch/Router can use only one queuing policy at a time. The policy is used on the entire GIGAswitch/Router. The default queuing policy is strict priority.

Configuring Layer-2 QoS

QoS policies applied to layer-2 flows allow you to assign priorities based on source and destination MAC addresses. A QoS policy set for a layer-2 flow allows you to classify the priority of traffic from:

- A specific source MAC address to a specific destination MAC address (use only when the port is in flow bridging mode)
- Any source MAC address to a specific destination MAC address

Before applying a QoS policy to a layer-2 flow, you must first determine whether a port is in address-bridging mode or flow-bridging mode. If a port operates in address-bridging mode (default) then you can specify the priority based on the destination MAC address and a VLAN ID. You can also specify a list of ports to apply the policy.

If a port operates in flow-bridging mode, the user can be more specific and configure priorities for frames that match both a source AND a destination MAC address and a VLAN ID. You can also specify a list of ports to apply the policy.

Note: In flow mode, you can also ignore the source MAC address and configure the priority based on the destination MAC address only.

When applying QoS to a layer-2 flow, priority can be assigned as follows:

- The frame gets assigned a priority within the switch. Select “low, medium, high or control”.
- The frame gets assigned a priority within the switch, AND if the exit ports are trunk ports, the frame is assigned an 802.1Q priority. Select a number from 0 to 7. The mapping of 802.1Q to internal priorities is the following: (0 = low) (1,2,3 =medium) (4,5,6 = high) (7 = control)

To set a QoS policy on a layer-2 flow, enter the following command in Configure mode:

Set a layer-2 QoS policy.	**qos set l2 name** *<name>* **source-mac** *<MACaddr>* **dest-mac** *<MACaddr>* **vlan** *<vlanID>* **in-port-list** *<port-list>* **priority control|high|medium|low|***<trunk-priority>*

Configuring Layer-3 and 4 QoS

QoS policies applied at layer-3 and 4 allow you to assign priorities based on specific fields in the IP and IPX headers. You can set QoS policies for IP flows based on source IP address, destination IP address, source TCP/UDP port, destination TCP/UDP port, type of service (TOS) and transport protocol (TCP or UCP). You can set QoS policies for IPX flows based on source network, source node, destination network, destination node, source port and destination port. A QoS policy set on an IP or IPX flow allows you to classify the priority of traffic based on:

- Layer-3 source-destination flows
- Layer-4 source-destination flows
- Layer-4 application flows

Configuring IP QoS Policies

To configure an IP QoS policy, perform the following tasks:

1. Identify the Layer-3 or 4 flow and set the IP QoS policy.

2. Specify the precedence for the fields within an IP flow.

Setting an IP QoS Policy

To set a QoS policy on an IP traffic flow, enter the following command in Configure mode:

Set an IP QoS policy.	**qos set ip** *<name> <priority> <srcaddr/mask>*|**any** *<dstaddr/mask>*|**any** *<srcport>*|**any** *<dstport>*|**any** *<tos>*|**any** *<interface-list>*|**any** *<protocol>*

Specifying Precedence for an IP QoS Policy

To specify the precedence for an IP QoS policy, enter the following command in Configure mode:

Specify precedence for an IP QoS policy.	**qos precedence ip [sip** *<num>*] **[dip** *<num>*] **[srcport** *<num>*] **[destport** *<num>*] **[tos** *<num>*] **[protocol** *<num>*] **[intf** *<num>*]

Configuring IPX QoS Policies

To configure an IPX QoS policy, perform the following tasks:

1. Identify the Layer-3 or 4 flow and set the IPX QoS policy.
2. Specify the precedence for the fields within an IPX flow.

Setting an IPX QoS Policy

To set a QoS policy on an IPX traffic flow, enter the following command in Configure mode:

Set an IPX QoS policy.	**qos set ipx** *<name> <priority> <srcnet>*|**any** *<srcmask>*|**any** *<srcport>*|**any** *<dstnet>*|**any** *<dstmask>*|**any** *<dstport>*|**any** *<interface-list>*|**any**

Specifying Precedence for an IPX QoS Policy

To specify the precedence for an IPX QoS policy, enter the following command in Configure mode:

Specify precedence for an IPX QoS policy.	**qos precedence ipx [srcnet** *<num>***]** **[srcnode** *<num>***] [srcport** *<num>***]** **[dstnet** *<num>***] [dstnode** *<num>***]** **[dstport** *<num>***] [intf** *<num>***]**

Configuring GIGAswitch/Router Queuing Policy

The GIGAswitch/Router queuing policy is set on a system-wide basis. The GIGAswitch/Router default queuing policy is strict priority. To change the queuing policy to weighted-fair queuing on the GIGAswitch/Router, enter the following command in Configure mode:

Set queuing policy to weighted-fair	**qos set queuing-policy weighted-fair**

If you want to revert the GIGAswitch/Router queuing policy from weighted-fair to strict priority (default), enter the following command in Configure mode:

Revert the GIGAswitch/Router queuing policy to strict priority.	**negate** *<line within active-configuration containing qos set queuing-policy weighted-fair>*

Allocating Bandwidth for a Weighted-Fair Queuing Policy

If you enable the weighted-fair queuing policy on the GIGAswitch/Router, you can allocate bandwidth for the queues on the GIGAswitch/Router. To allocate bandwidth for each GIGAswitch/Router queue, enter the following command in Configure mode:

Allocate bandwidth for a weighted-fair queuing policy.	**qos set weighted-fair control** *<percentage>* **high** *<percentage>* **medium** *<percentage>* **low** *<percentage>*

Monitoring QoS

The GIGAswitch/Router provides display of QoS statistics and configurations contained in the GIGAswitch/Router.

To display QoS information, enter the following command in Enable mode:

Show all IP QoS flows	**qos show ip**
Show all IPX QoS flows.	**qos show ipx**
Show all L2 QoS flows.	**qos show l2 all-destination all-flow ports** *<port-list>* **vlan** *<vlanID>* **source-mac** *<MACaddr>* **dest-mac** *<MACaddr>*

Chapter 11 Performance Monitoring Guide

Performance Monitoring Overview

The DIGITAL GIGAswitch/Router is a full wire-speed layer-2, 3 and 4 switching router. As packets enter the GIGAswitch/Router, layer-2, 3, and 4 flow tables are populated on each line card. The flow tables contain information on performance statistics and traffic forwarding. Thus the GIGAswitch/Router provides the capability to monitor performance at Layer 2, 3 and 4. Layer-2 performance information can be accessed by SNMP through MIB-II and by the command line through the *l2-tables* command. Layer-3 and 4 performance statistics can be accessed by SNMP through RMON/RMON2 and through the CLI through the *statistics show* command. In addition to the monitoring commands listed, you can find more monitoring commands listed in each chapter of the DIGITAL *GIGAswitch/Router User Reference Manual*.

To access statistics on the GIGAswitch/Router, enter the following commands in Enable mode:

Show DVMRP routes.	**`dvmrp show routes`**
Show HTTP statistics.	**`http show statistics`**
Show all TCP/UDP connections and services.	**`ip show connections`**
Show all IP routes.	**`ip show routes`**
Show all IPX routes.	**`ipx show tables routing`**
Show all MAC addresses currently in the L2 tables.	**`l2-tables show all-macs`**
Show info about MACs residing in a port's L2 table.	**`l2-tables show port-macs`** *`<port-list>`*
Show all L2 flows (for ports in flow-bridging mode.	**`l2-tables show all-flows`**
Show information about the master MAC table.	**`l2-tables show mac-table-stats`**
Show information about a particular MAC address.	**`l2-tables show mac`**

Show info about multicasts registered by IGMP.	**`l2-tables show igmp-mcast-registrations`**
Show whether IGMP is on or off on a VLAN.	**`l2-tables show vlan-igmp-status`**
Show info about MACs registered by the system.	**`l2-tables show bridge-management`**
Show SNMP statistics.	**`snmp show statistics`**
Show ICMP statistics.	**`statistics show icmp`**
Show IP interface's statistics.	**`statistics show ip`**
Show unicast routing statistics.	**`statistics show ip-routing`**
Show IPX statistics.	**`statistics show ipx`**
Show IPX interface's statistics.	**`statistics show ipx-interface`**
Show IPX routing statistics.	**`statistics show ipx-routing`**
Show multicast statistics.	**`statistics show multicast`**
Show port error statistics.	**`statistics show port-errors`**
Show port normal statistics.	**`statistics show port-stats`**
Show RMON statistics.	**`statistics show rmon`**
Show traffic summary statistics.	**`statistics show summary-stats`**
Show TCP statistics.	**`statistics show tcp`**
Show UDP statistics.	**`statistics show udp`**
Show TACACS server statistics.	**`tacacs show stats`**
Show all VLANs.	**`vlan list`**